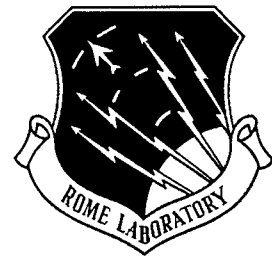


RL-TR-96-192, Volume I (of two)
Final Technical Report
September 1996



METAMODELING TECHNIQUES AND APPLICATIONS

Mission Research Corporation

Dr. Don Caughlin

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

19970305 039

DTIC QUALITY INSPECTED 3

**Rome Laboratory
Air Force Materiel Command
Griffiss Air Force Base, New York**

This report has been reviewed by the Rome Laboratory Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RL-TR-96-192, Vol I (of two) has been reviewed and is approved for publication.

APPROVED: 

ALEX F. SISTI
Project Engineer

FOR THE COMMANDER: 

JOSEPH CAMERA
Technical Director
Intelligence & Reconnaissance Directorate

If your address has changed or if you wish to be removed from the Rome Laboratory mailing list, or if the addressee is no longer employed by your organization, please notify RL/IRAE, 32 Hangar Road, Rome, NY 13441-4114. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document require that it be returned.

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE September 1996	3. REPORT TYPE AND DATES COVERED Final May 94-Jan 96		
4. TITLE AND SUBTITLE METAMODELING TECHNIQUES AND APPLICATIONS		5. FUNDING NUMBERS C: F30602-94-C-0110 PE: 62702F PR: 4594 TA: 15 WU: K0		
6. AUTHOR(S) Dr. Don Caughlin				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Mission Research Corporation 102 South Tejon, Suite 200 Colorado Springs, CO 80903-2200		8. PERFORMING ORGANIZATION REPORT NUMBER		
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Rome Laboratory/IRAE 32 Hangar Road Rome, NY 13441-4114		10. SPONSORING/MONITORING AGENCY REPORT NUMBER RL-TR-96-192 Vol I (of two)		
11. SUPPLEMENTARY NOTES Rome Laboratory Project Engineer: Alex F. Sisti/IRAE/(315) 330-4518				
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited		12b. DISTRIBUTION CODE		
13. ABSTRACT (Maximum 200 words) Tactical simulation models used by the Department of Defense to assess the capabilities of combat systems and tactics are highly complex. It is often difficult to determine the relationship of individual factors to the performance of the modeled process. A technique called metamodeling has the ability to address this problem. This research developed the capability to build dynamic metamodels of tactical engagement simulations.				
14. SUBJECT TERMS Metamodeling Reduced Order Modeling Model Abstraction		15. NUMBER OF PAGES 324		
		16. PRICE CODE		
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT Same as Report	

EXECUTIVE SUMMARY

Statement of the Problem. Tactical simulation models used by the Department of Defense to assess the capabilities of combat systems and tactics are highly complex. It is often difficult to determine the relationship of individual factors to the performance of the modeled process. Consequently, it is not easy to use the results of the model in another simulation or couple multiple models to investigate a larger issue. The result is a proliferation of point-designed models and simulations, expensive upgrade and maintenance, and our inability to efficiently answer many of the more difficult questions raised by the acquisition and operational communities.

Background. A technique called metamodeling has the ability to address this problem. Metamodeling is model abstraction technique that projects the simulation model onto a reduced order subspace defined by new constraints or regions of interest. Use of metamodels, however, was restricted to static models that only represented the input-output behavior of simulations.

Results. This research developed the capability to build dynamic metamodels of tactical engagement simulations. Dynamic metamodels can accurately model the simulated processes. Going beyond an input-output map, dynamic metamodeling can facilitate software reuse, large scale model integration, verification, and validation. The ability to develop dynamic metamodels was the result of a new approach supported by a taxonomy of metamodeling problems, solution structures, and metamodeling methods.

The theoretical approach used for the model description in this research is significantly different from the usual approaches followed by either the operations research or engineering communities. The framework centered on the behavior of the system, the behavioral equations that specify the behavior, and latent (unobserved) variables which may be present from first principles.

The taxonomy of simulations defined classes of metamodeling problems which could be used to specify which metamodeling method was most appropriate. This process is also supported by a new taxonomy of metamodel structures and methods to generate the metamodel. The new classification of methods and structures allowed the separation of the standard metamodeling process into a few well-defined steps. The first eight steps of the procedure became the foundation for the "problem definition;" the remaining steps were grouped in an iterative scheme as the "metamodeling process." By restructuring the 13 step procedure, the research directly coupled the *a priori* knowledge to the structure of the metamodel and simplified the development process.

Recommendations. While these new methods work well, significant expertise is required in system identification, the design of experiments, and statistics. Additional research is needed to build a robust system that will support the subject matter expert. This system will assist the analyst who is not familiar with model abstraction techniques but needs to reuse a piece of code, integrate different models, or verify a new version of a simulation.

TABLE OF CONTENTS

VOLUME I Final Report

Introduction

1. Chapter Outline	1-1
2. Statement of the Problem	1-1
3. Research Purpose	1-2
4. Organization of the Report	1-3
5. References	1-5

Metamodels

1. Chapter Outline	2-1
2. Introduction	2-1
3. The Modeling Process	2-2
4. Metamodeling	2-5
5. Metamodel Procedures	2-10
6. Metamodeling Research	2-11
7. References	2-14

Framework

1. Chapter Outline	3-1
2. Introduction	3-1
3. Requirements for Representations	3-7
4. Identifiability	3-10
5. Representing Discrete Event Systems	3-12
6. References	3-14

Air Force Metamodeling Problems

1. Chapter Outline	4-1
2. Introduction	4-3
3. Metamodel Requirements (Purpose of the Metamodel)	4-4
4. Simulation Characteristics	4-10
5. Wargaming and Simulation Matrix	4-22
6. Metamodeling Classes	4-24
7. References	4-28

Metamodel Structures

1. Chapter Outline	5-1
2. Introduction	5-2
3. System Description (Representation)	5-3
4. Predictor Models	5-6
5. Probabilistic Models	5-15
6. References	5-30

Metamodeling Methods	
1. Chapter Outline	6-1
2. Introduction.....	6-3
3. Prediction Error Methods (PEM).....	6-15
4. Correlation Approaches	6-28
5. Maximum Likelihood Approaches.....	6-30
6. Optimization.....	6-39
7. Approximation Techniques for Identification.....	6-43
8. Relationships Between Methods.....	6-50
9. References	6-52
Determination of Model Structure and Order	
1. Chapter Outline	7-1
2. Introduction.....	7-2
3. General.....	7-3
4. Model Structure Selection	7-12
5. Generalized Model Order Determination.....	7-15
6. Method Based Model Order Determination.....	7-21
7. Model Order Reduction Methods.....	7-23
8. References	7-28
Determination of Model Validity	
1. Chapter Outline	8-1
2. Introduction.....	8-2
3. General.....	8-3
4. Measures of Local Validity	8-5
5. Measures of Global Validity.....	8-11
6. Block Structured, Norm-Bounded Uncertainty.....	8-21
7. References	8-22
Experimental Design	
1. Chapter Outline	9-1
2. Introduction.....	9-2
3. Statistical Experimental Design	9-9
4. Experimental Design for Identification	9-21
5. Input-Output Variable Selection for Metamodeling	9-29
6. References	9-32
Metamodeling Combat Simulations	
1. Chapter Outline	10-1
2. Introduction.....	10-2
3. General Approach.....	10-5
4. Metamodeling Procedures.....	10-12
5. Procedural Notes	10-30
6. References	10-31

Additional Research Issues	
1. Chapter Outline	11-1
2. Introduction.....	11-1
3. Theoretical Aspects	11-1
4. System Development	11-4
5. References	11-6
Summary of Results	
1. Chapter Outline	12-1
2. Research Summary	12-1
3. Results and Discussion.....	12-6
Conclusions and Recommendations	
1. Conclusions	13-1
2. Recommendations.....	13-1
APPENDIX 1 Simulation in the Air Force	
1. Chapter Outline	A1-1
2. Introduction.....	A1-1
3. Requirement	A1-1
4. Capability Provided.....	A1-2
5. Levels of Analysis	A1-2
6. Use of Simulation in the Air Force	A1-3
7. Modeling and Simulation in the Acquisition Process.....	A1-5
8. Modeling and Simulation in the Operational Environment	A1-9
9. References	A1-10
APPENDIX 2 Hierarchical Simulation	
1. Chapter Outline	A2-1
2. Introduction.....	A2-1
3. Hierarchical Construction	A2-2
4. Application to Metamodeling.....	A2-3
5. References	A2-4
APPENDIX 3 Adaptive Simulative Annealing	
1. Chapter Outline	A3-1
2. Introduction.....	A3-1
3. Metropolis Algorithm	A3-2
4. Statistical Mechanics.....	A3-3
5. Information Theory.....	A3-7
6. Simulated Annealing	A3-8
7. Summary	A3-12
8. References	A3-13
VOLUME II Research Summaries	
Analysis of TERSM Model 7	1-1
ASA Least Squares Metamodel	2-1
TERSM Output Error Metamodel	3-1
ASA Stochastic Markov Metamodel.....	4-1
SIMTAX Feature Space and Results.....	5-1
Model Order Selection.....	6-1

LIST OF FIGURES

VOLUME I

Figure 2.4.1.	Direct Model Correspondence.....	2-6
Figure 2.4.2.	Inverse Model Correspondence	2-7
Figure 2.5.1.	Iterative Metamodeling Process.....	2-11
Figure 4.4.1.	Sampled-Data Continuous System Model.....	4-18
Figure 4.6.2.	Distance of Each Simulation from the Center of its Category.....	4-26
Figure 4.6.3.	Distance of All Simulations from the First Cluster Centroid	4-26
Figure 4.6.4.	Distance of Each Simulation from the Center of its Category.....	4-27
Figure 6.2.1.	Experimental Format for the Equation Error Method.....	6-8
Figure 6.2.2.	Experimental Format for the Output Error Method.....	6-9
Figure 6.2.3.	Experimental Format for the Prediction Error Method.....	6-9
Figure 7.3.1	A Comparison of Two Linear Approximations	7-4
Figure 8.5.1.	White Noise Sequence.....	8-14
Figure 8.5.2.	White Noise Power Spectral Density	8-14
Figure 8.5.3.	White Noise Autocorrelation Sequence	8-15
Figure 9.2.1.	General Time Varying Linear System Description.....	9-3
Figure 9.2.2.	Closed-Loop Experimental Setup	9-7
Figure 10.4.1	Metamodeling Approach	10-12
Figure 10.4.2	Analysis of a Simulation for an Analytical Metamodel.....	10-14
Figure 10.4.3	Analysis of a Simulation for an Simulation Metamodel.....	10-15
Figure 10.4.4	Analysis External Simulation Features	10-16
Figure 10.4.5	Selection of Metamodel Components	10-17
Figure 10.4.6	Determination of System Description.	10-18
Figure 10.4.7	Determination of System Class	10-18
Figure 10.4.8	Metamodel Structure for Static Systems.....	10-19
Figure 10.4.9	Metamodel Structure for Linear Time-Invariant Systems.	10-20
Figure 10.4.10	Metamodel Structure for Time Varying Systems.....	10-20
Figure 10.4.11	Metamodel Structure for Nonlinear Systems.....	10-21
Figure 10.4.12	Selection of the Form of the Identifier and Error Criterion.....	10-21
Figure 10.4.13	Main ID Method Selection Routine	10-22
Figure 10.4.14	Selection of Prediction Error Methods.....	10-23
Figure 10.4.15	Selection of Maximum Likelihood Methods.....	10-23
Figure 10.4.16	Selection of Approximation Methods	10-24
Figure 10.4.17	Selection of Optimization Methods.....	10-24
Figure 10.4.18	Selection of Simulation Input Values	10-25
Figure 10.4.19	Data Acquisition Sequence.....	10-26
Figure 10.4.20	Metamodel Input Data Analysis Sequence	10-27
Figure 10.4.21	Generating the Metamodel	10-28

VOLUME II

Figure 1.2.1.	Variance Decomposition Matrix.	1-3
Figure 1.2.2.	Condition Indices for the Singular Values.....	1-4
Figure 1.2.3.	Variance Proportion of Condition Number 1929.46.....	1-5
Figure 1.2.4.	Variance Proportion of Condition Number 1504.83.....	1-5
Figure 1.2.5.	Variance Proportion of Condition Numbers 146.069 through 195.468.....	1-6
Figure 1.2.6.	Variance Proportion of Condition Numbers 66.6565 through 119.17.....	1-6
Figure 1.2.7.	Residual Power Spectrum using the Full Data Set.....	1-12
Figure 1.2.8.	Residual Autocorrelation using the Full Data Set.....	1-12
Figure 1.2.9.	Residual Cumulative Probability Distribution using the Full Data Set.	1-13
Figure 1.2.10	Residual Power Spectrum Excluding Variates 20, 21,22.....	1-16
Figure 1.2.11.	Residual Autocorrelation Excluding Variates 20, 21,22.	1-16
Figure 1.2.12.	Residual Cumulative Probability Distribution Excluding Variates 20, 21,22.....	1-17
Figure 1.2.13.	Residual Power Spectrum with Variate 22 Removed.	1-20
Figure 1.2.14.	Residual Autocorrelation with Variate 22 Removed.....	1-20
Figure 1.2.15.	Residual Cumulative Probability Distribution with Variate 22 Removed.....	1-21
Figure 1.2.16.	Residual Power Spectrum with Variate 21 Removed.	1-24
Figure 1.2.17.	Residual Autocorrelation with Variate 21 Removed.....	1-24
Figure 1.2.18.	Residual Cumulative Probability Distribution with Variate 21 Removed.....	1-25
Figure 1.2.19.	Residual Power Spectrum with Variate 20 Removed.	1-28
Figure 1.2.20.	Residual Autocorrelation with Variate 20 Removed.....	1-28
Figure 1.2.21.	Residual Cumulative Probability Distribution with Variate 20 Removed.....	1-28
Figure 2.2.1.	Cost Function for the Static Least Squares Metamodel.	2-3
Figure 3.1.1.	TERSM Functional Flow.....	3-1
Figure 3.1.2.	TERSM Emitter Field and Aircraft Path.	3-2
Figure 3.1.3	Actual Data and Metamodel Estimates for the Last 976 Data Points.....	3-7
Figure 4.3.1.	Comparison of Actual and Modeled Data.	4-2
Figure 5.3.1.	Paradox Query.	5-8
Figure 5.5.1.	Sorted distance from the Cluster Centroid.	5-11
Figure 5.5.2.	Distance of all simulations from the first the Cluster Centroid.	5-11
Figure 5.5.3.	Distance of all simulations from all Cluster Centroids.	5-12
Figure 6.2.1	Phase Plane Plot of X2 versus X1.....	6-3
Figure 6.2.2	Phase Plane Plot of X3 versus X1.....	6-3
Figure 6.2.3	Phase Plane Plot of X3 versus X2.....	6-4
Figure 6.2.4	Data Used for Determination of Model Order.....	6-4
Figure 6.2.5.	A Comparison of Results With Prior Results.	6-6

LIST OF TABLES

VOLUME I

Table 2.3.1.	Model Development Methods	2-2
Table 2.3.2.	Model Forms.....	2-2
Table 2.3.3.	Levels of Detail, Characteristics, and Simulation Modes	2-3
Table 4.3.1.	Metamodel Purpose Summary.....	4-9
Table 4.3.2.	Metamodel Purpose Summary.....	4-9
Table 4.4.1.	Selections for Analytical Simulations.....	4-11
Table 4.4.2.	Selections for Training and Education Simulations.....	4-11
Table 4.4.3.	Selections for the Domain	4-12
Table 4.4.4.	Selections for the Span.....	4-12
Table 4.4.5.	Environment.....	4-13
Table 4.4.6.	Selections for Force Composition.....	4-14
Table 4.4.7.	Category of Weapons or Systems Simulated.....	4-14
Table 4.4.8.	Selections for Mission Area.....	4-15
Table 4.4.9.	Human Participation.....	4-16
Table 4.4.10.	Selections for Time Processing.....	4-16
Table 4.4.11.	Treatment of Randomness.....	4-17
Table 4.4.12.	Sidedness	4-17
Table 4.4.13.	Definition of Basis for Internal Processing.....	4-19
Table 4.4.14.	Definition of Process Description for Internal Processing.....	4-19
Table 4.4.15.	Options for System, Input, and Output Processing	4-20
Table 4.4.16.	Options for Selection of the Internal Process	4-20
Table 4.4.17.	Metamodel Purpose Summary.....	4-21
Table 4.5.1.	Wargaming and Simulation Matrix	4-22
Table 5.3.1.	Parameter Descriptions and Model Representations.....	5-4
Table 5.3.2.	System Descriptions	5-4
Table 5.3.3.	System Classes and Representations	5-5
Table 5.4.1.	Static System Model Structures.....	5-6
Table 5.4.2.	Linear Time-Invariant (or Stationary) System Model Structures	5-7
Table 5.4.3.	Linear Nonstationary and Time-Varying and Nonlinear System Model Structures.....	5-8
Table 5.4.4.	NonLinear System Model Structures	5-9
Table 6.2.1.	Notation.....	6-4
Table 6.2.2.	Model Structures and Limitations.....	6-5
Table 6.2.3.	Open Loop and Off-Line Identification Methods	6-6
Table 6.2.4.	On-Line and Closed-Loop Identification Methods	6-7
Table 7.3.1.	Model Structure and Order Determination Methods	7-11
Table 8.3.1.	Error Sources.....	8-4
Table 8.5.1.	Information Based Validation Methods.....	8-10
Table 8.5.2.	Data Accuracy Model Validation Methods	8-10
Table 8.5.3.	Critical Values for the Test Statistic for the K-S Test	8-17
Table 8.5.4.	Values for K in the K-S Test	8-18
Table 10.3.1	Metamodeling Approach	10-5

VOLUME II

Table 1.2.1.	Results of the Analysis of the Condition Indices.....	1-4
Table 1.2.2.	Results of the Analysis of the pi Matrix.	1-7
Table 2.2.1.	Input-Output Data for Construction of the Static Least Squares Metamodel.....	2-2
Table 5.1.1.	Catalog categories selected for SIMTAX.	5-1
Table 5.2.1.	SIMTAX Database Structure.	5-3
Table 5.2.2.	Model Type Database.....	5-4
Table 5.2.3.	Domain Database.	5-4
Table 5.2.4.	Span Database.....	5-4
Table 5.2.5.	Environment Database.....	5-5
Table 5.2.6.	Force Composition Database.....	5-6
Table 5.2.7.	Scope of Conflict Database.	5-6
Table 5.2.8.	Time Processing Database.	5-6
Table 5.2.9.	Human Participation Database.....	5-7
Table 5.2.10.	Treatment of Randomness Database.	5-7
Table 5.2.11.	Sidedness Database.	5-7
Table 6.2.1	Comparison of Correlations for Orthogonal and Stochastically Independent CVA.	6-5

CHAPTER 1

INTRODUCTION

1. CHAPTER OUTLINE

1. Chapter Outline.....	1-1
2. Statement of the Problem.....	1-1
3. Research Purpose.....	1-2
3.1. General Approach.....	1-2
3.2. Background.....	1-3
3.3. Knowledge-Based Support -- Expert Systems.....	1-3
4. Organization of the Report.....	1-3
5. References	1-5

2. STATEMENT OF THE PROBLEM

Tactical simulation models used by the Department of Defense to assess the capabilities of combat systems and tactics are highly complex. It is often difficult to determine the relationship of individual factors to the performance of the modeled process [1]. Consequently, it is not easy to use the results of the model in another simulation or couple multiple models to investigate a larger issue. The result is a proliferation of point-designed models and simulations, expensive upgrade and maintenance, and our inability to efficiently answer many of the more difficult questions raised by the acquisition and operational communities [2].

A technique called metamodeling has the ability to facilitate this type of assessment. As an abstraction,¹ a metamodel is a projection of the model onto a subspace defined by new constraints or regions of interest. Selection of the parameters used for the projection (the construction of a metamodel) involves: *a priori* knowledge; the data; a set of metamodel structures; and rules to determine the best model to realize the data. This research addresses the issues associated with metamodeling tactical simulations.

Since metamodeling is a developing field, there was no unifying theory or set procedure for constructing a metamodel. **This research contributed to a unifying theory.** The research also provided the analyst with methods to generate accurate models that meet the needs of the decision maker.

¹Model abstraction refers to the process of hiding the implementation details of an object from the users of that object.

3. RESEARCH PURPOSE

The purpose of this research was twofold:

- First, this research advanced the ability to develop reliable, consistent, and usable relationships via metamodeling.
- Second, this research applied metamodeling to develop models required to address current Air Force issues.

It should be noted that as a method of model abstraction, metamodeling can be applied to many classes of models. This research, however, focused on metamodels of combat simulations.

An additional goal was to provide procedures, algorithms, and a knowledge base to assist in metamodeling. Therefore, in addition to specific solutions, MRC provided methodologies for optimizing metamodels that can be used by an analyst and were capable of transition to automated modeling systems.

3.1. General Approach

This research built upon an existing metamodeling procedure outlined in [1]:

1. Determine the purpose of the metamodel
2. Identify the response
3. Identify important response characteristics
4. Identify input factors
5. Identify important input characteristics
6. Specify the experimental region
7. Select validity measures
8. Specify required validity
9. Postulate a metamodel based on:
 - Input - Output response characteristics
 - Experimental region dimensions
 - Required validity
10. Select an experimental design
11. Obtain data
12. Fit the metamodel
13. Assess the validity of the model

The first eight procedural steps provide the prior knowledge for the selection and fit of a model to meet the needs of the analyst. The remaining steps define a process that will actually determine the metamodel.

There are many techniques available to attack an Air Force metamodeling problem. **The major problem with metamodeling as a discipline is the lack of connectivity between the problem structure (prior knowledge) and Steps 9, 10, and 13.** The issue was the proper use of systems theory and experimental design to arrive at the "best" metamodel

structure that solves a particular problem. **A solid connection between the problem (prior knowledge) and solution technique was needed.**

Objective 1 of the research addressed the first eight steps and provided the requirements background to develop the connection between the metamodeling problem and the solution structure.

Objective 2 addressed steps 9, 10 and 13 to advance the field by postulating a theoretical base to connect the metamodel structure to the intended application. The result is a cluster of metamodel structures and problems and a correspondence between the two.

3.2. Background

Metamodeling research in the operations research (OR) community has been able to define the nature of the problem. The control engineering (CE) community has developed system theory and identification techniques to improve its ability to predict and control dynamical systems. This research combined results from these disciplines and provides a connection between the two that is directly applicable to combat simulations.

3.3. Knowledge Base Support -- Expert Systems

Advances in object oriented programming techniques combined with progress in understanding the collection, processing, storing, accessing, and use of knowledge have fostered the development of useful expert systems. "Intelligent software" is more flexible than conventional software. It can respond in more complex ways and can deliver highly tailored recommendations. The systems can provide multiple answers with different degrees of certainty and thereby provide the non-expert user with multiple options.

An expert system is the union of declarative knowledge and inference. The knowledge base contains the declarative knowledge. The inference engine controls the application of that knowledge. It is an algorithm that dynamically directs or controls the system when it searches the knowledge base. Expert systems speed up human work by at least an order of magnitude by placing expert knowledge and systematic search and reasoning skills at the fingertips of the average analyst.

Development of a knowledge base for metamodeling is not a primary focus of this research. The research, however, will result in information that could be used to support an expert system. This effort will collate and provide lessons learned in a suitable format for inclusion in an expert system at a later date.

4. ORGANIZATION OF THE REPORT

The Final Report for Air Force Rome Laboratory Contract F30602-94-C-0110, "Modeling Techniques and Applications," is contained in two volumes. Volume I contains the theoretical foundation, the results of the experiments, and procedures developed to metamodel combat simulations. Volume II contains the results of the individual experiments that were undertaken to answer critical questions raised by theoretical issues.

Volume I is organized as follows:

- | | |
|------------|---|
| Chapter 1 | Introduces the research and the report; |
| Chapter 2 | Provides a more detailed discussion of modeling and metamodels; |
| Chapter 3 | Provides the theoretical framework that underlies the research; |
| Chapter 4 | Addresses Objective 1 and defines the elements of the metamodeling problems that were the focus of this effort; |
| Chapter 5 | Begins Objective 2 and provides a summary of the structures that are available for the metamodels; |
| Chapter 6 | Provides a compilation of the many methods that are available to actually develop metamodels; |
| Chapter 7 | Discusses techniques to determine model order; |
| Chapter 8 | Covers methods to address the validity of the model; |
| Chapter 9 | Discusses issues associated with experimental design and the acquisition of data for the metamodel; |
| Chapter 10 | Provides the detailed procedures to follow in order to metamodel combat simulations; |
| Chapter 11 | Contains a discussion of some of the remaining research issues; |
| Chapter 12 | Provides a summary of research results; |
| Chapter 13 | Contains conclusions reached by the effort. |

Volume II is not designed to be a coherent discussion of research results; it is a supplement to Volume I and provides the details of experiments that resulted in the procedures and results presented in that volume. Volume II begins with an analysis of a least squares model of the Tactical Electronic Reconnaissance Simulation Model. Chapter 2 presents a least-squares model that resulted from optimization using Adaptive Simulated Annealing (ASA). Chapter 3 is an output error model generated using the framework and approach outlined in Volume I, Chapter 3. Chapter 4 demonstrates a stochastic model identified using ASA. Chapter 5 contains the details of the metamodeling problem space discussed in Volume I, Chapter 4. Chapter 6 contains experiments to demonstrate model set and order selection.

5. REFERENCES

1. M. A. Zeimer, et al, "Metamodel Procedures for Air Engagement Simulation Models," *IRAE Technical Report*, January 1993.
2. A. F. Sisti, "Large-Scale Battlefield Simulation Using a Multi-Level Model Integration Methodology," *IRAE Technical Report RL-TR-92-69*, April 1992.

CHAPTER 2

METAMODELS

1. CHAPTER OUTLINE

1. Chapter Outline.....	2-1
2. Introduction.....	2-1
3. The Modeling Process.....	2-2
3.1. General.....	2-2
3.2. Hierarchical Modular Modeling/Knowledge Representation.....	2-3
3.2.1. General Structure.....	2-3
3.2.2. Synthesis of Models.....	2-4
4. Metamodeling.....	2-5
4.1. Introduction.....	2-5
4.2. Metamodeling Techniques.....	2-5
4.3. Metamodeling Objectives.....	2-8
4.4. Uses of Metamodels Derived From Inverse Modeling.....	2-8
4.5. Limitations.....	2-9
5. Metamodel Procedures.....	2-10
5.1. Sequential and Iterative Procedures.....	2-10
6. Metamodeling Research.....	2-11
6.1. General Approach.....	2-11
6.2. Objective 1.....	2-12
6.3. Objective 2.....	2-13
7. References.....	2-14

2. INTRODUCTION

This chapter provides an introduction to the research effort, defining the problem and the general research approach.

Specifically this chapter covers the modeling process to include simulation and hierarchical modular modeling. Next, we introduce metamodels, metamodeling techniques, the selected method from the inverse problem, and the objectives of the mathematical process used to generate the metamodels. Also, we cover the uses and limitations of metamodels generated from data. Finally, we introduce general metamodeling procedures and the research required to implement these procedures. We conclude by outlining the specific research objectives.

3. THE MODELING PROCESS

3.1. General

A model is a structure that can be used for understanding the behavior of a system [1]. It is one method of expressing a theory. The model can be a physical structure such as a wind tunnel model used to determine the aerodynamics of an aircraft, or it can be a conceptual model represented by interactions, a system of equations, or a simulation.

Models are arrived at by a number of methods and have several forms. Table 2.3.1 outlines the methods used to develop a model and Table 2.3.2 depicts several of the forms that they may take.

Table 2.3.1. Model Development Methods.

DERIVATION	CHARACTERISTICS
Descriptive	Attempt to describe an observed regularity without seeking explanation
Prescriptive	Normative => implies the establishment of standards
Inductive	Inference of a general model from observations of particular instances
Deductive	Reasoning from known to unknown the mathematical model analytically and use experimental observations to fill in the gaps
Analog Models	Analogy: Resemblance between attributes, circumstances or effects Useful in imitating (not duplicating) a system

Table 2.3.2. Model Forms.

FORM	BASIS	CHARACTERISTICS
Physical	Direct	Characteristics of the system are preserved
	Indirect	Only a mathematical similarity between the system and model
Mathematical	Laws	Set of equations Parameters - Numerical values
		Conservation and continuity
Simulation		
	Structure	Interconnected components

The final form of a model was a simulation. Mathematical simulation (as opposed to a more general definition of simulation where physical models or environments are used to represent the behavior) is a particular type of model structure that is a procedure for selecting an arbitrary element of the model behavior and defining an algorithm for computing it. Since the simulation model is the focus of this research, we will discuss it further.

The simulation model is the foundation of what is called Modeling and Simulation (M&S). Being arbitrary, there are multiple levels of detail possible in the M&S environment. Hierarchical modeling techniques have been developed to support these different levels of representation. It should be noted that there is no right or wrong level; the selection of the level of detail is a function of the fidelity or confidence required by the analyst.

Table 2.3.3. Levels of Detail, Characteristics, and Simulation Modes.

LEVEL OF DETAIL	CHARACTERISTICS	SIMULATION MODE
Coarse representation Grossly aggregate model	Concerned with system performance Reduces processing requirements Lacks intricate detail	Fast model mode
Intermediate representation	Model isolated viewpoints Selected entities / areas of interest	Metamodel mode
Detailed representation Model every entity	Incorporates detail into scenario Cost, time and resources - greatly stressed Completely detailed analysis	Robust mode

3.2. Hierarchical Modular Modeling/Knowledge Representation

3.2.1. General Structure

Assume that we are given two models. If the model description is in the proper form, then we can create a new model by specifying how the input and output ports are connected. This allows modules (models) to be connected by an operation called **coupling** [2]. If A and B are coupled together, then we have a new model, AB, which is a coupled model which is once again in a modular form. In this sense, modularity means the description of a model in such a way that it has a recognized input and output through which all interaction is accomplished. The ability to couple the models is called closure under coupling, and it enables the hierarchical construction of models.

Elements of model bases that are closed under coupling consist of both atomic and coupled models, each of which is called a component. Each atomic model has three parts to its description:

1. The input-output specification giving the input and output ports their ranges,
2. Static structure giving the state and auxiliary variables and ranges,
3. Dynamic structure which provides the external and internal transition specification.

A coupled model has a different description:

1. The input-output specification,
2. Names of the components (other coupled or atomic models) that are coupled together,
3. Coupling specification.

While modular discrete event models still require specification of inputs and outputs, they must accommodate the fact that events determine the dynamics of the models.

Hierarchical construction, made possible by the successive coupling of larger and larger components, goes beyond standard object oriented programming. Model descriptions must be converted into a class specification. A class specification is a template for generating identical instances of the same model along with a convention for naming the different instances of the same model. The structure and components of a hierarchical, modular model are portrayed by a composition tree. Generalizing the composition to represent a family of models results in a system entity structure. In this structure, there can be several possible models to represent it. The decomposition of the structure is an aspect since there may be several possible decompositions for a given entity.

The entity structure/model base combination provides a unifying description of knowledge consistent with system theoretic insights. System theory distinguishes between structure (constitution of the system) and behavior (outer manifestation). Knowledge is represented in the decomposition, coupling, and taxonomies (class definitions). Behaviors, causal relationships, are integrated into the models.

3.2.2. Synthesis of Models

The entity structure and the model base combine to facilitate model construction. The model base contains files for the various model classes. Model construction consists of two passes. First there is a top-down pruning of the entity structure to identify the desired components in the model base. This is followed by a bottom-up synthesis to construct the new model. Elements of selected classes are coupled together following the coupling specification.

4. METAMODELING

4.1. Introduction

Assume that we have a model of a system that cannot be used directly. A solution may not exist; it may be too complicated for a closed-form solution; it may require too much time to numerically determine a particular solution; or it may be a high-fidelity simulation that provides much more detail than we are interested in. Efficient use of this model requires a "black-box" approximation of the causal time dependent behavior of the model -- a metamodel.

A metamodel is a mathematical approximation of the system relationships defined by another, more detailed model (in our case -- a tactical simulation). It is black-box approximation of the casual time dependent behavior of a simulation model that allows the assessment of individual factors on the performance of the simulation model. These approximations can be used for: studying system behavior; predicting responses; sensitivity analysis; or optimization.

Furthermore, metamodeling can be used in the synthesis of complex simulations from lower level components [2]. The best way to model a large scale, complex software system is to model different portions of the system at different levels of detail and interconnect the elements [3]. Then, by selecting the appropriate representations (metamodels) for each of the components, the combined simulation can be executed at the proper level of fidelity. When needed; e.g., to obtain specific data or verify overall simulation fidelity, it would be possible to "zoom" in on certain elements (execute a component model with higher fidelity).

Metamodeling can model a single realization of a simulation, multiple realizations with different initial conditions, or a Monte-Carlo ensemble of the same initial conditions.

Metamodeling is poised for growth and is on the verge of providing the Air Force a significant increase in capability. This increase in capability will have major impacts on every aspect of the Air Force mission from combat decision support to the large scale integration of complex simulations.

4.2. Metamodeling Techniques

There are two basic techniques available for metamodeling: direct and inverse modeling.

First, a metamodel could be developed by applying basic principles to generate a more abstract (approximate) version of the original model. This would be an example of direct modeling. Direct modeling is characterized by a specification of the elements of the model. Complicated systems are modeled by "tearing" a system into its components, modeling these components in a process called "zooming," and then interconnecting these components to construct a "physical" realization of the system [3,4,5]. The level of abstraction is controlled by the detail of the specification. The model reveals the structure of the theory and allows the prediction of the response to exogenous inputs as a function

of the state of the system. The solution of this modeling problem requires an understanding of the process being modeled and methods to express this understanding.

Metamodels developed using this technique are "stand alone" versions. The relationship between the real system, the original model, and the metamodel is contained in the two mappings from the underlying system to each of the models. Figure 2.4.1 depicts this correspondence.

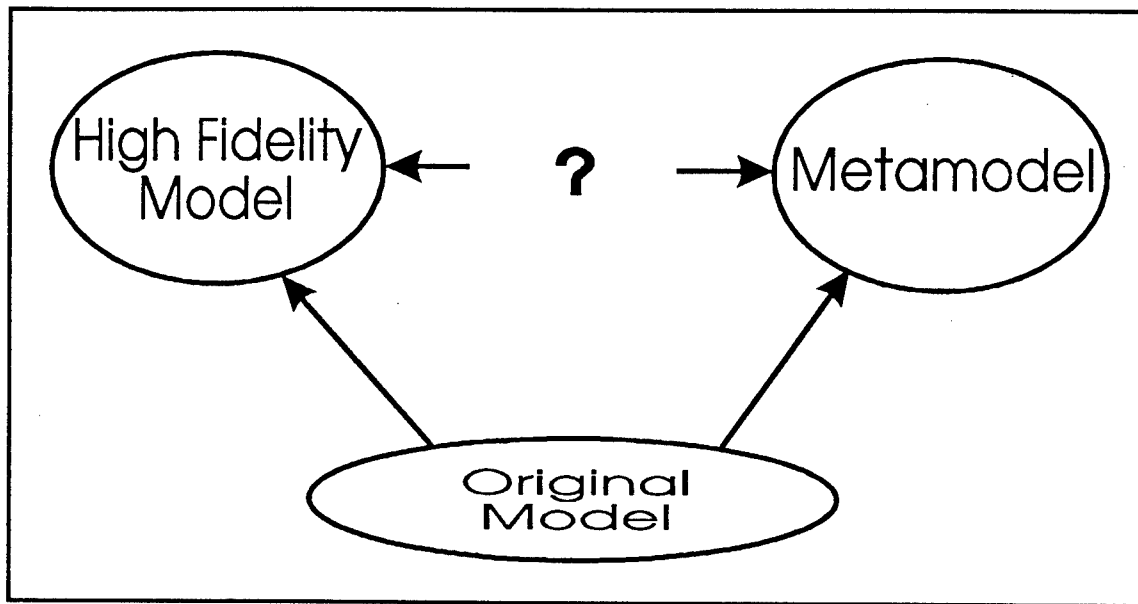


Figure 2.4.1. Direct Model Correspondence.

As seen from the figure, there is no guarantee that a usable correspondence will exist between the metamodel and the model [6,7]. Traceability from the high-fidelity model to the more abstract, lower fidelity metamodel becomes a significant issue. Also, this technique still requires an *a priori* understanding of the structure of the elements and the interconnections between these elements at the specific level of fidelity selected. This, in fact, could be a difficult and risky task, and lack of this knowledge is often the reason that a high fidelity simulation was used in the first place.

The second technique develops the metamodel from the input-output data generated by the original model or simulation. This technique is an example of the "inverse problem," and is represented by Figure 2.4.2. From the figure, we see that the correspondence between the model and the metamodel is direct. The issues now are the level of fidelity, range of applicability, and accuracy of the response. These are a function of the metamodeling technique and data.

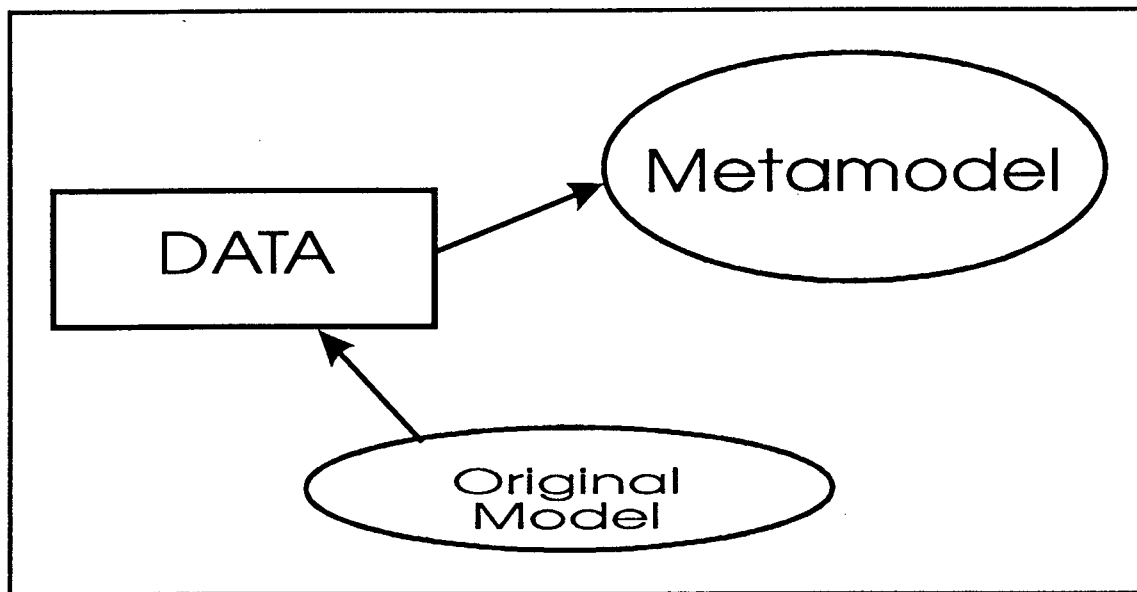


Figure 2.4.2. Inverse Model Correspondence.

As difficult as the direct modeling problem may be, the inverse problem is much more complex. In this case, we have some estimate (measure) of the input and output response but do not have a complete characterization of the process by which the outputs are generated. While a properly posed direct problem generally has a solution, the inverse problem usually has multiple solutions out of which an acceptable solution (if it exists) must be selected. This technique explicitly results in a mathematical approximation between the inputs and responses - this is the technique we consider.

It should be noted that there is a significant difference between our approach and much of the prior research. Most of the previous work that could be categorized as metamodeling consisted of procedures to determine the best polynomial fit to a set of input-output data. The researchers concentrated on the statistical properties of the data. In our approach, we are not trying to fit data! We are attempting to identify the underlying processes that define the system that generated the data (or, in our terminology, the behavior). The focus is not on statistics but on the system theoretic properties of the manifest behavior.

The techniques that will be used for the system identification are generally considered to be elements of "Intelligent Control." In intelligent control there are three coupled variations of the estimation problem. The first issue is to identify the structure and the values of the parameters that define the mathematical model. This is the parameter estimation problem. Once the mathematical model is defined, we then use this model to estimate the value of the variables by making measurements of the system. This is the variable or state estimation problem. The estimation problem arises because the measurements are corrupted by both measurement noise and system disturbances not deterministically accounted for by the model.

However, we are never absolutely certain that the structure or parameterization of the model represents the underlying system. Consequently, with each estimate of the variables an assessment must be made concerning the validity of the model (i.e., is the error

predicted by the model consistent with the observation). This is a combined state and parameter estimation or adaptive estimation problem. Each technique will be used in metamodeling combat simulations.

4.3. Metamodelling Objectives

A metamodelling procedure should provide (1) parameters, (2) error estimates on the parameters, and (3) a measure of the goodness of fit. In this section, we will concentrate on the first requirement -- the parameters, and the requirements necessary to be able to generate appropriate relationships.

Given a phenomenon that we would like to describe, we desire a mathematical expression as the model. Assume that this phenomenon produces **outcomes** that are elements of a set U . A model for this phenomenon will probably generate certain of these outcomes and exclude others. Consequently, the outcomes recognized by the model, B , are a subset of the universal set U , and are called the **behavior** of the model. For the inverse modeling problem, we define a model class \mathcal{M} with elements $M = (U, B)$ where $B_f \subseteq U$ is the behavior of M .

We shall see later that the behavior allowed by the metamodel is usually contained within the set of behaviors allowed by the model. Yet the metamodel cannot be more powerful than the model. The difference is in the data set D . The data set for the original model is not contained with the set of behaviors allowed by the metamodel.

For the inverse modeling problem, we define a model class \mathcal{M} . From an experiment, we obtain data from measurements. Different realizations of the attributes of the phenomenon may result in the same data, or they may lead to different observed data caused by the interference of other phenomenon or latent variables. To proceed, we will assume that the data consist of observed realizations of the phenomenon itself.

Important considerations in the selection of a modeling procedure are falsification and the notion of a more powerful model. The more a model forbids (at the level of accuracy we desire), the "better" it is. A model is unfalsified by the data if $D \in U$ and $D \in B$. A model (U, B_1) is more powerful than (U, B_2) if $B_1 \subseteq B_2$.

The objective is to determine the Most Powerful Unfalsified Model (MPUM). A model is the MPUM based on the data D if: (1) $M \in \mathcal{M}$; (2) \mathcal{M} is unfalsified by D ; and (3) M is more powerful than any other model satisfying (1) and (2). The MPUM may not exist. If the MPUM does exist, it is unique.

4.4. Uses of Metamodels Derived From Inverse Modeling

Properly developed, a metamodel derived from inverse modeling is clearly a mathematical approximation between a set of input factors and responses generated by the high fidelity model. As such, it allows the assessment of individual factors on the performance of the simulation and can directly be used to study system behavior, predict responses, perform sensitivity analysis, or optimize elements of the system to meet requirements.

Since the traceability to the high fidelity model is immediate, metamodeling can be used in the synthesis of complex simulations from lower level components[2]. An efficient method for modeling a large scale, complex software system is to model the different portions of the system at multiple levels of detail, and interconnect the elements [2,4]. Then, by selecting the appropriate representations (metamodels) for each of the components, the combined simulation can be executed at the proper level of fidelity. When needed; e.g., to obtain specific data or verify overall simulation fidelity, it would be possible to "zoom" in on certain elements (execute a component model with higher fidelity) [5].

4.5. Limitations

As opposed to direct modeling where the components are synthesized by tearing and zooming, the inverse modeling problem obtains the model from the data. Models and laws obtained in this way should be considered as descriptive, not necessarily interpretive models.

Care must be taken in the setup of the metamodeling problem. It is possible to correlate two variables when there is no logical or mathematical reason to believe that such a relationship exists.

The experimental design must provide input-output sequences that correctly represent the system structure. When the metamodel is determined, it is not possible to ask "What is the probability that a particular set of fitted parameters is correct?" because there is no statistical universe of models from which the correct one is chosen. There is just one model and a statistical universe of data sets that are drawn from it. It is possible to ask, however, "Given a particular set of parameters, what is the probability that this data set could have occurred?" We can identify the probability of the data given the parameters as the likelihood of the parameters given the data [8].

In addition to the problem setup and experimental design, the metamodel solution comes with limits of its own. Using the space spanned by the original model as the full order model, the metamodel is a reduced order approximation. This reduction inherently limits the span of the manifest (exogenous) variables associated with the behavior (input or output, if such a map exists). Consequently, the behaviors allowed by the metamodel will exist within a subspace of the original model.

Assuming that an input-output map exists for the model, input values will be restricted to a domain within which the metamodel will be applicable. Outside of this hypersurface, application of the metamodel may provide numbers but will not generate an output that is representative of the modeled system. Also, assuming appropriate inputs, the output of the metamodel can only be guaranteed to be approximately correct. As a projection, the metamodel will not contain all of the detail of the original model. There are output error bounds that are a function of both the metamodel and the input.

Given that it is possible to determine the MPUM, the next issue is determination of the system from the MPUM. This is the issue of identifiability [9]. In order to address issues

associated with metamodels obtained from inverse modeling, we must establish a framework that defines a system model and incorporates the representation of that model. Given this framework, there are results which guarantee that any unstructured input will be sufficiently rich to identify a controllable system. This research will focus on requirements to identify systems from simulations.

5. METAMODEL PROCEDURES

5.1. Sequential and Iterative Procedures

As a starting point, the metamodeling procedure is defined as [10]:

1. Determine the purpose of the metamodel
2. Identify the response
3. Identify important response characteristics
4. Identify input factors
5. Identify important input characteristics
6. Specify the experimental region
7. Select validity measures
8. Specify required validity
9. Postulate a metamodel based on:
 - Input - Output response characteristics
 - Experimental region dimensions
 - Required validity
10. Select an experimental design
11. Obtain data
12. Fit the metamodel
13. Assess the validity of the model

The first eight procedures provide the prior knowledge for the selection and fit of a model to meet the needs of the analyst. The remaining procedures are implemented in a recursive fashion as shown in Figure 2.5.1 [11].

While the recursive steps in Figure 2.5.1 apply to the metamodel procedures directly, an explanation of the additional detail follows. Step 9 is the initial selection of the community design, model set, and criterion of fit as shown in Figure 2.5.1. Step 12 is the calculation of the model based on selections and data from prior steps. The other steps in the metamodel procedure (Steps 10, 11 and 13) are shown explicitly in Figure 2.5.1.

Step 11 is an application of the state estimation (filtering) problem, and Step 12 is concerned with computational numerics. Although important, they were not central to the success of metamodeling. Existing techniques were used for these steps.

Steps 9, 10 and 13 are the most critical. The major problem with metamodeling as a discipline is the lack of connectivity between the problem structure (prior knowledge) and Steps 9, 10, and 13. These steps were the focus of the second objective.

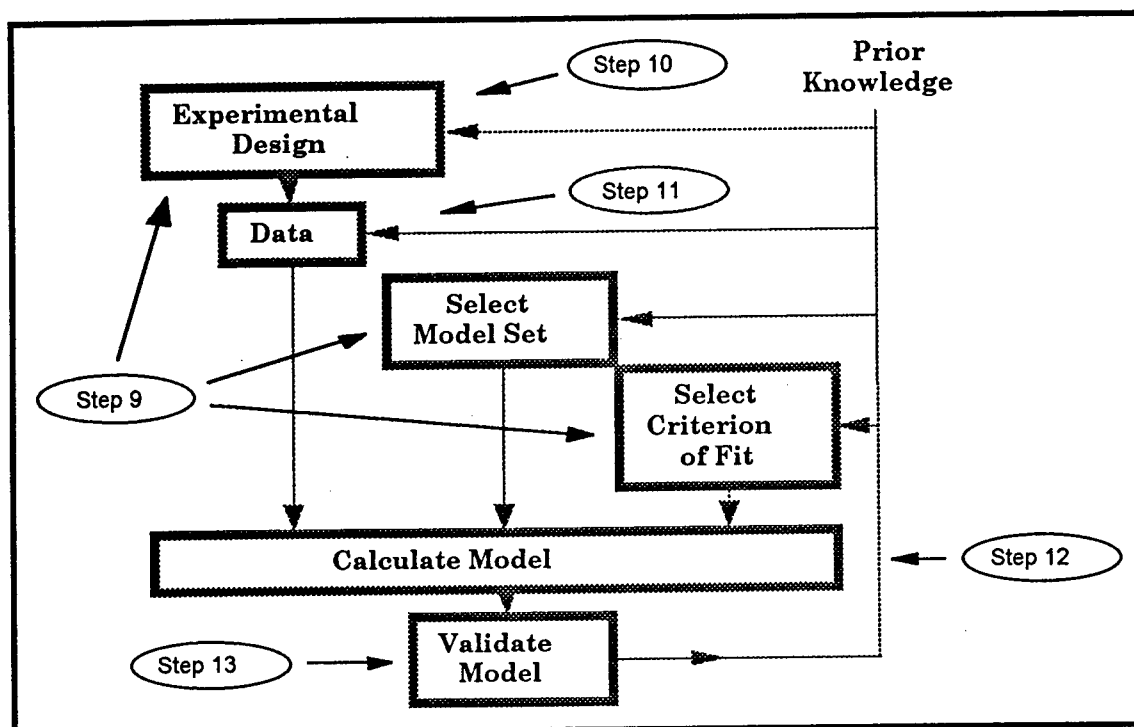


Figure 2.5.1. Iterative Metamodeling Process.

6. METAMODELING RESEARCH

6.1. General Approach

The first eight procedural steps provide the prior knowledge for the selection and fit of a model to meet the needs of the analyst. The remaining steps define a process that will actually determine the metamodel.

There are many techniques available to attack an Air Force metamodeling problem. The issue was the proper use of systems theory and experimental design to arrive at the "best" metamodel structure that solves a particular problem. **A solid connection between the problem (prior knowledge) and solution technique was needed and provided by this research.**

We define a metamodeling problem as the direct sum of the model (simulation) and metamodel requirements. The first objective of the research addressed the first eight steps and provided the requirements background to define the metamodeling problem.

The second objective concentrated on steps 9, 10 and 13 to postulate a theoretical base to connect a specific metamodel structure to the metamodeling problem (intended application). The result was clusters of metamodel structures, problems, and a correspondence between the two.

The objectives of this research are to:

1. Define classes of Air Force metamodeling problems based on the simulations and *a priori* knowledge (metamodel use). Determine criteria for clustering metamodeling problems. Apply these criteria to selected simulations.
2. Categorize the set of available metamodel structures and determine criteria for application to Air Force metamodeling problems. Demonstrate use of these criteria.

Objective 1 provides the background to address this connection between the problem and the structure.

Objective 2 addresses the primary issues posed in this research. This part of the research will concentrate on steps 9, 10 and 13 to advance the field by postulating a theoretical base to connect the metamodel structure to the intended application. The result will be a cluster of metamodel structures, problems, and a correspondence between the two.

6.2. Objective 1

The first objective of this research focused on the steps that provide the prior knowledge (the first eight steps):

1. Determine the purpose of the metamodel
2. Identify the response
3. Identify important response characteristics
4. Identify input factors
5. Identify important input characteristics
6. Specify the experimental region
7. Select validity measures
8. Specify required validity

The connection between prior knowledge and the metamodeling technique began with an analysis of the types of problems facing the Air Force analyst and engineer. In order to group these problems for use with an identification technique, significant characteristics of these problems were identified. These characteristics defined classes of Air Force metamodeling problems based on the simulations and *a priori* knowledge (metamodel use).

A broad look at the "universe" of simulations was required to define general classes of metamodeling problems. With the insight provided by the theory of dynamical systems, MRC reviewed existing simulations and identified the metamodeling problems (prior knowledge and characteristics) associated with as many simulations as possible.

Selected models of interest to Rome Laboratory (RL) were further evaluated. Based on the range of applications of the metamodel (prior knowledge), feature vectors were determined that were used to define a space of metamodeling problems.

Once the feature space that encompasses the selected metamodeling problems was defined, the next step was to determine classes of metamodeling problems. This was accomplished by evaluating the density of the metamodeling problems in the feature space

and, based on the characteristics of the space, selecting criteria for clustering. Classes of metamodeling problems were then defined by these clusters.

Having identified classes of metamodeling problems, representative problems from the different classes were selected. These representative problems were used as the basis for further metamodeling research. Results of this research should apply to the entire class.

6.3. Objective 2

Research for the second objective addressed the steps that defined and determined the metamodel (Steps 9 through 13):

9. Postulate a metamodel based on:
 - Input - output response characteristics
 - Experimental region dimensions
 - Required validity
10. Select an experimental design
11. Obtain data
12. Fit the metamodel
13. Assess the validity of the model

For this objective we categorized the set of available metamodel structures and determined criteria for application to Air Force metamodeling problems. Once developed, use of these criteria were demonstrated.

7. REFERENCES

1. V. Vemuri, *Modeling of Complex Systems*, Academic Press 1978.
2. B. P. Zeigler, "Hierarchical Modular Modeling/Knowledge Representation," *Proc. 1986 Winter Simulation Conf.*, pp 120-137, 1986.
3. A. F. Sisti, "Large-Scale Battlefield Simulation Using a Multi-Level Model Integration Methodology," *RL-TR-92-69*, April 1992.
4. J. C. Willems, "Paradigms and Puzzles in the Theory of Dynamical Systems," *IEEE Trans. on Automat. Contr.*, vol 36, no. 3, pp 259-294, March 1991.
5. A. F. Sisti, "A Model Integration Approach to Electronic Combat Effectiveness Evaluation," *RL-TR-89-183*, October 1989.
6. A. W. Naylor, and G. R. Sell. *Linear Operator Theory in Engineering and Science*, New York: Springer Verlag, 1982.
7. H. L. Royden, *Real Analysis*, Macmillan Publishing Company, 1988.
8. Press, et al, *Numerical Recipes*, Cambridge University Press, 1986.
9. H. Kwakernaak and R. Sivan, *Linear Optimal Control Systems*, Wiley-Interscience, 1972.
10. M. A. Zeimer, et al, "Metamodel Procedures for Air Engagement Simulation Models," *IRAE Technical Report*, January 1993.
11. L. Ljung, *System Identification: Theory for the User*, Prentice-Hall, New Jersey, 1987.

CHAPTER 3

FRAMEWORK

1. CHAPTER OUTLINE

1. Chapter Outline	3-1
2. Introduction.....	3-1
2.1. Background	3-2
2.2. Definitions for General System Models.....	3-2
2.3. Dynamical Systems.....	3-3
2.4. Representations.....	3-4
2.5. Controllability and Observability.....	3-6
2.6. Discrete-Event Systems (DES).....	3-6
2.7. Modeling.....	3-7
3. Requirements for Representations	3-7
4. Identifiability.....	3-10
4.1. Definitions.....	3-10
4.2. Discussion.....	3-10
5. Representing Discrete Event Systems.....	3-12
5.1. Simulation.....	3-12
5.2. Discrete Event Nature of Modeled Events.....	3-13
6. References	3-14

2. INTRODUCTION

The theoretical background used for the model description in this proposed research is significantly different from the usual approaches followed by either the Operations Research (OR - analysis) or engineering communities.

The following discussion outlines a framework for the identification of dynamical systems. The framework centers on the behavior of the system, the behavioral equations that specify the behavior, and latent variables which may be present from first principles. This structure and definitions follow the presentation given in [1].

This theory of dynamical systems begins with the essence of the system and not with a structure and assumptions that facilitate a solution technique. Consequently, this theory provides a basis that includes all of the issues associated with modeling and modeling from data.

2.1. Background

Given a phenomenon that we would like to describe, we desire a mathematical expression as the model [1]. Assume that this phenomenon produces **outcomes** that are elements of a set U . A model for this phenomenon will probably generate certain of these outcomes and exclude others. Consequently, the outcomes recognized by the model B , are a subset of the universal set U , and are called the **behavior** of the model. For the inverse modeling problem, we define a model class \mathcal{M} with elements $M=(U,B)$ where $B \subseteq U$ is the behavior of M .

2.2. Definitions for General System Models

We have defined the behavior of the model as the outcomes recognized by the model -- B (a subset of the universal set U). Therefore, define a mathematical model as the pair (U,B) with U the universe of outcomes produced by the underlying phenomenon, and B , the behavior of the model. Often, the behavior of the model is described by a set of equations that leads to a behavioral equation representation of the pair (U,B) . To accommodate this, consider an abstract set, E , called the equating space, and $f_1, f_2: U \rightarrow E$. With this space, and the functions f_1, f_2 , the behavioral representation for the model becomes (U,E,f_1,f_2) .

The behavior of the mathematical model (U,B) can be a set of equilibrium conditions so that $B = \{u \in U \mid f_1(u) = f_2(u)\}$ or it can be a set of inequalities where $B = \{u \in U \mid f_1(u) \leq f_2(u)\}$. Although the equations uniquely specify the behavior of the model, the converse is not true. Two distinct behavioral equations can represent the same behavior. The important result of the modeling procedure is the behavior, the solution set of the behavioral equations, not the behavioral equations themselves.

A mathematical model is **linear** if U is a vector space and B is a linear subspace of U . Assume that $U = I \times O$, where I is the input space, O is the output space, and B is the graph of a system map $F: I \times O$ called an I/O map. These assumptions allow an **input-output model** where $(U,B) \Leftrightarrow (I \times O, B) \Leftrightarrow (I, O, F)$. If the past does not contain any information about the future other than the information in the behavioral relationships, the map is **nonanticipating**. The relationship between input and output may or may not be nonanticipating. If the map is nonanticipating, an I/O map interprets the attributes I as causing the output O , and can be described by the behavioral equation $y = F(u)$. However, this approach does not make an *a priori* distinction between inputs and outputs of the model. Given a mathematical model, the choice of input and output should be deduced from the model, not imposed upon it.

In summary, the modeling procedure requires that we specify the variables that we want to model (specify the universal set U), and then identify the possible outcomes in the behavior. Often, however, we will require additional variables in addition to those we seek to model. These other variables are called **latent variables**. These variables are required whenever we develop a metamodel by the method of tearing, where the system is

viewed as the interconnection of subsystems. Consequently, we expand the mathematical model to allow latent variables by defining a triple (U, L, B_f) . Here L is the set of latent variables, $B_f \subseteq U \times L$, with $B_f \equiv \{u \in U \mid \exists l \in L \text{ such that } (u, l) \in B_f\}$.

Our last topic in the discussion of general system models is the concrete description of the model. Let \mathbf{M} be the set of mathematical models. Each element $M \in \mathbf{M}$ denotes a mathematical model (U, B) . The model set of interest may be uncountable. The idea then, is to parameterize M and perform the search over the parameter set. A **parametrization** of M consists of a model structure which is a set \mathcal{D}_m and a surjective¹ map $\mathcal{M}: \mathcal{D}_m \rightarrow \mathcal{M}(\theta)$. The set \mathcal{D}_m is the parameter space with $\theta \in \mathcal{D}_m$ determining the behavioral equations.

2.3. Dynamical Systems

Again, the model for a dynamical system is defined in terms of its behavior. A dynamical system is a family of trajectories without reference to I/O maps, variables, or behavioral equations. The system is coupled to its environment and is not defined by a model associated with it. A model for a dynamical system Σ is simply a triple $\Sigma = (T, W, B)$ with $T \subseteq \mathbf{R}$ the time axis, W the signal space, and $B \subseteq W^T$ the behavior -- the set of all maps from T to W , a family of W -valued time trajectories.

The behavioral equations, such as difference or differential equations, lead to **representations** of dynamical systems. A dynamical system is linear if W is a vector space (over a field F^2) and B is a linear subspace of W^T . A dynamical system $\Sigma = (T, W, B)$ is said to be **time invariant** if $\sigma^t B = B$ for all $t \in T$. Here σ^t is the time-shift operator: $(\sigma^t f)(t') \triangleq f(t'+t)$.

A dynamical system $\Sigma = (T, W, B)$ is said to be **complete** if $\{w \in B\} \Leftrightarrow \{w|_{[t_1, t_2]} \in B|_{[t_1, t_2]}, \forall t_1, t_2 \in T, t_1 \leq t_2\}$. Completeness is an important property affecting the mathematical structure that defines the behavioral equations that

¹**Surjective:** For $f: X \rightarrow Y$, the domain of f is the set X . The range of f is the set of values taken by f , the set $\{y \in Y: (\exists x) [y = f(x)]\}$. If the range of f is Y then f is a function **onto** Y : f is surjective.

² Any set that satisfies the following for all real numbers x, y , and z :

- A1. $x + y = y + x$.
- A2. $(x + y) + z = x + (y + z)$.
- A3. $\exists 0 \in \mathbf{R}$ such that $x + 0 = x$ for all $x \in \mathbf{R}$.
- A4. For each $x \in \mathbf{R}$ there is a $\omega \in \mathbf{R}$ such that $x + \omega = 0$.
- A5. $xy = yx$.
- A6. $(xy)z = x(yz)$.
- A7. $\exists 1 \in \mathbf{R}$ such that $1 \neq 0$ and $x \cdot 1 = x$ for all $x \in \mathbf{R}$.
- A8. For each x in \mathbf{R} different from 0 there is $\omega \in \mathbf{R}$ such that $x\omega = 1$.
- A9. $x(y + z) = xy + xz$.

represent dynamical systems. Simply put, for a complete system, the way a signal behaves at $t = \pm \infty$ is of no consequence as to whether or not the signal obeys the laws of the system.

Dynamical systems acquire their importance from the fact that they exhibit memory or the potential to model phenomena where the past influences the future. In this context, a dynamical system is said to have a finite **memory span** Δ ($\Delta \in \mathbf{T}$, $\Delta > 0$) if $w_1, w_2 \in B$, $w_1(t) = w_2(t)$ for $0 \leq t \leq \Delta \Rightarrow \{w_1 \wedge w_2 \in B\}$ ³. Where

$$(w_1 \wedge w_2)(t) = \begin{cases} w_1(t) & \text{for } t < 0 \\ w_2(t) & \text{for } t \geq 0 \end{cases}$$

If $\Delta = 0$, the dynamical system is **memoryless**; if $\Delta = 1$ (in discrete time) the system is **Markovian**. Therefore, for a system with a finite memory span, the past is independent of the future. Σ is **Δ complete** ($\Delta \in \mathbf{T}$, $\Delta \geq 0$) if $\{w \in B\} \Leftrightarrow \{(\sigma^t w)|_{[0, \Delta]} \in B|_{[0, \Delta]} \forall t \in \mathbf{T}\}$.

A dynamical system with latent variables can also be defined as an extension of a dynamical system with only manifest variables. In this case the system is defined as $\Sigma = (\mathbf{T}, \mathbf{W}, \mathbf{L}, B_f)$. \mathbf{T} is the time axis, \mathbf{W} is the space of manifest (directly observable -- external) variables, \mathbf{L} the space of latent variables, and $B_f \in \mathbf{W} \times \mathbf{L}$ is the full behavior. Also, consistent with the general model discussed above, an **input/output dynamical system** can be defined if: (1) the input itself cannot be explained by the model, and (2) once the model is understood, the input is given and the initial conditions set, the output is uniquely defined.

Having defined the system by it's behavior, the structure of behavioral equations are contained in representations of the system. We now cover some specific representations of dynamical systems.

2.4. Representations

The model is defined by the behavior that it allows. The behavior can be defined by a set of inequalities or equations. The structure of the equations is a representation of the model. Recall that the objective of metamodeling is to determine the MPUM. Given a data set \mathcal{D}_m , and a model set $\mathcal{M}(\theta)$, with θ the particular parameter vector, we will use the following definition to quantify the concept of the model structure:

A model structure \mathcal{M} is defined as a differentiable mapping from a connected open subset \mathcal{D}_m of \mathbf{R}^d to a model set $\mathcal{M}(\theta)$, such that the gradients of the predictor functions are stable.

³Here, \wedge denotes concatenation.

Given a model structure, consider the following behavioral representation obtained by assuming $W = R^q$, $E = R^g$, and f_1, f_2 linear (E is the equating space):

$$R_L w(t+L) + R_{L-1} w(t+L-1) + \dots + R_{l+1} w(t+l+1) + R_l w(t+l) = 0$$

where $R_L, R_{L-1}, \dots, R_{l+1}, R_l \in R^{g \times q}$. If we introduce the polynomial matrix

$$R(s, s^{-1}) = R_L s^L + R_{L-1} s^{L-1} + \dots + R_{l+1} s^{l+1} + R_l s^l \in R^{g \times q}[s, s^{-1}]$$

the above system of equations can be written as $R(\sigma, \sigma^{-1})w = 0$. This representation is only a function of current and past signals (outputs) and is called an **autoregressive (AR) representation**.

If the system that we are trying to model suggests latent variables to describe the behavior, the autoregressive representation can be expanded to include a moving average part of the past latent variables, resulting in an **autoregressive-moving-average (ARMA) representation**. In this case, the behavioral difference equations relate the time-series of the manifest variables $w: Z \rightarrow R^q$ to the time-series of the latent variables $l: Z \rightarrow R^q$. Let $R(s, s^{-1}) \in R^{g \times q}[s, s^{-1}]$ and $M(s, s^{-1}) \in R^{g \times d}[s, s^{-1}]$ and define the ARMA system as:

$$R(\sigma, \sigma^{-1})w = M(\sigma, \sigma^{-1})l.$$

These equations represent a dynamical system with latent variables, $\Sigma = (Z, R^q, R^d, B_f)$. An important class of ARMA systems are those where $R(s, s^{-1}) = I$. This yields a **moving average (MA) representation**: $w = M(\sigma, \sigma^{-1})l$.

As we have seen, latent variables form an important part of representing systems in that they provide a way of formalizing models that contain auxiliary variables. One method of representing latent variables is through state variables. A state-space dynamical system is defined as a dynamical system with latent variables, $\Sigma = (T, W, X, B_s)$ with $X \subseteq L$, such that the full behavior $B_s \in W \times X$ satisfies the axiom of state. In this case the latent variables, the states, contain sufficient information about the past so as to determine future autonomous behavior.

We can combine the above constructs to define a class of models with all of the advantages of completeness, described by the difference (differential) equation; state form, the memory is displayed through the latent variables; and nonanticipating input-output, an explicit cause and effect structure. This representation is an input/state/output representation and is the model class most amenable to analysis, synthesis and simulation. This system is defined as the quintuple $\Sigma = (T, U, Y, X, B_s)$ where U is the input signal space, Y the output signal space, X the state space.

2.5. Controllability and Observability

All dynamical systems are not controllable. In a controllable system, the past trajectory does not have a lasting influence on the far future. Sooner or later, any other trajectory within the controllable subspace, can be attained. In an autonomous system, the past trajectory determines its future completely. Consequently, the lack of controllability implies predictability. As we develop the capability to better understand and control our environments, our ability to predict that environment can suffer. We are limited in our ability to predict by our ability to observe.

Let $\Sigma = (T, W, B)$ be a time-invariant dynamical system. Σ is said to be controllable for all $w_1, w_2 \in B$ if there exists a $t \in T$, $t \geq 0$, and a $w: T \cap [0, t] \rightarrow W$ such that $w' \in B$, with $w': T \rightarrow W$ defined by

$$(w')(t) = \begin{cases} w_1(t') & \text{for } t' < 0 \\ w(t') & \text{for } 0 \leq t' \leq t \\ w_2(t' - t) & \text{for } t' \geq t \end{cases}$$

While controllability is intrinsic to the dynamic system, observability is also a function of the representation of that system. This comes about because observability is only an issue for dynamical system model **representations** that have latent variables (by definition, if the variable is a manifest variable it is observed), and is a property where an unobserved signal can be deduced from one which is observed.

2.6. Discrete-Event Systems (DES)

The above framework is consistent with the formalized discrete-event systems in theoretical computer science. The behavior is similar to the formal language; a state-space system is like an automation; latent variables are replaced by production rules; interconnections are communications. The most significant difference is the lack of behavioral models (equations) in the theory of DES. Also, completeness is usually violated in a DES by initiation and termination rules for event strings.

Since the DES is not complete, representation of these systems requires special consideration. We will see that while completeness is required to represent a dynamical system by a behavioral difference equation, results for representation of complete systems may be generalized to a class on noncomplete systems that meet specific restrictions.

A linear time-invariant dynamical system (Z, R^q, B) is called an l_2 -system if B is a linear shift-invariant closed subspace of $l_2(Z; R^q)$ ⁴. Define $\overline{B^{pc}}$ as the closure of B with respect

⁴ $l_2(Z; R^q)$ is the set of infinite sequences such that $\sum_{k=1}^{\infty} |x_k|^2 < \infty$. See [2,3] for details.

to the topology of pointwise convergence. With these definitions, results for complete systems may be generalized to l_2 -systems satisfying $B = \overline{B}^{\text{pc}} \cap l_2(Z; R^q)$.

2.7. Modeling

Interconnecting is particularly useful in modeling. Direct modeling usually begins with a system which we view as an interconnection of a family of component subsystems. Via a process called tearing, we zoom in on the individual subsystems and set up mathematical models and interconnections for each of the subsystems. Together the interconnected system provides a model for the overall system.

Synthesis of a mathematical model can be obtained by interconnecting, in a preassigned way, standardized building blocks called modules. Four devices, a delay, an amplifier with a gain K , an adder, and a fork (a two-output device with each output equal to the input) can be combined to define three dynamical systems. These systems are the module stack that defined the I/O dynamical system behavior; a wiring diagram that defines an I/O dynamical system that equates subsystem variables; and an I/O interface with input and output terminals. Any dynamical system that belongs to \mathcal{L}^q can be synthesized in this manner.

3. REQUIREMENTS FOR REPRESENTATIONS

With a framework established to characterize system models, we now address the key issue of the inverse modeling problem: "What properties of the behavior allow the system to be represented by a difference (or differential) equation of a particular type?" Analysis of these properties will result in rules and constraints for the setup and design of metamodels.

To begin with, we must establish the most basic system properties. These are the properties that allow the description of the behavior via an equality. From [1] we have the following proposition for discrete-time systems:

Proposition I. Let $\Sigma = (Z, W, B)$ be a discrete time dynamical system. The following conditions are equivalent:

1. Σ is time-invariant, complete, and has memory span L ;
2. Σ is time-invariant and L -complete;
3. Σ can be described by a behavioral difference equation with lag L .

Proof. See Proposition 1.1 \Rightarrow J. C. Willems, "Models for Dynamics," Dynamics Reported, vol. 2, pp. 171-269, 1989.

Therefore, for a system to be represented by means of a difference equation, it has to be complete (it cannot have initialization or termination conditions at $t = \mp \infty$) with a finite

memory span so that observation of a trajectory on a finite time interval allows conclusions about past behavior independent of what will happen in the future.

In addition to the conditions required to represent a dynamical system as a difference equation (Proposition I), the restriction of the behavior to an autoregressive representation (where $B = \{w: Z \rightarrow R^q | R(\sigma, \sigma^{-1})w = 0\}$) adds the following equivalencies to Proposition I for this representation.

Theorem II. Let $\Sigma = (Z, R^q, B)$ be a dynamical system. Then, in addition to Proposition I, the following are equivalent:

1. $\exists R(s, s^{-1}) \in R^{* \times q}[s, s^{-1}]$ such that $B = \ker R(\sigma, \sigma^{-1})$;
2. $B \in \mathcal{L}^q$ where \mathcal{L}^q is the family of all linear shift-invariant closed subspaces of \mathcal{L} .

Proof. See Propositions 4.1A & 4.2 \Rightarrow J. C. Willems, "Models for Dynamics," Dynamics Reported, vol. 2, pp. 171-269, 1989.

Combining the results of Proposition I and Theorem II, we see that for a system to be described by AR-equations it must be linear, complete, and time invariant.

So far we have discussed systems with only manifest variables (by virtue of their AR representation). Now consider systems with latent variables that have an ARMA representation where the behavioral difference equations relate the time-series of the manifest variables w to the time-series of the latent variables l . We know that every system $\Sigma = (Z, R^q, R^d, B)$ can be described by an ARMA representation of behavioral equations and that the ARMA system induces a manifest behavior that satisfies the moving average constraints. If we begin with the same system restrictions required for an AR representation, with $B \in \mathcal{L}^{q+d}$ we can address the induced manifest behavior.

Theorem III: Let the dynamical system with latent variables $\Sigma_f = (Z, R^q, R^d, B_f)$ be linear, time invariant, and complete. Then the manifest system which it represents $\Sigma = (Z, R^q, B)$ is also linear time invariant and complete.

Proof. See Propositions 4.1C \Rightarrow J. C. Willems, "Models for Dynamics," Dynamics Reported, vol. 2, pp. 171-269, 1989.

Consequently, if the dynamical system with latent variables represented by the ARMA system is linear time-invariant and complete, then the latent variables could be completely eliminated from the equations resulting in an AR representation. The cost of eliminating the latent variable is an increase in the lag of the AR system.

From Theorem II (item 2), every behavior $B \in \mathcal{L}^q$ allows an AR representation. What restrictions must be placed on the system to allow a MA-representation? This question is addressed in the next theorem.

Theorem IV. The dynamical system $\Sigma = (\mathbf{Z}, \mathbf{R}^q, \mathcal{B})$, with $\mathcal{B} \in \mathcal{L}^q$ is controllable if and only if there exists $M(s, s^{-1}) \in \mathbf{R}^{q \times m}[s, s^{-1}]$ such that $\mathcal{B} = \text{im } M(\sigma, \sigma^{-1})$.

Proof. See Propositions 4.3 \Rightarrow J. C. Willems, "Models for Dynamics," Dynamics Reported, vol. 2, pp. 171-269, 1989.

From this theorem, we see that if the dynamical system is controllable (if it is possible to eventually steer the system to a desired trajectory) then the system will also allow an MA representation with $w = M(\sigma, \sigma^{-1})l$.

With the first three theorems we have defined the system properties necessary for the AR-, ARMA-, and MA- representations. The remainder of this section will consider the generalization of these representations into input-output and input/state/output structures.

Theorem V. Let $\Sigma = (\mathbf{Z}, \mathbf{R}^q, \mathcal{B})$, with $\mathcal{B} \in \mathcal{L}^q$. Then there exists a componentwise partition of $\mathbf{R}^q = \mathbf{R}^m \times \mathbf{R}^p$ such that the resulting $(\mathbf{Z}, \mathbf{R}^m, \mathbf{R}^p, \mathcal{B})$ defines a nonanticipating I/O system.

Proof. See Theorem 4.1 \Rightarrow J. C. Willems, "Models for Dynamics," Dynamics Reported, vol. 2, pp. 171-269, 1989.

Consequently, an input-output dynamical representation can be defined if and only if it can be described by an AR-system of behavioral equations $P(\sigma, \sigma^{-1})y = Q(\sigma, \sigma^{-1})u$ with $P(s, s^{-1}) \in \mathbf{R}^{p \times q}[s, s^{-1}]$, $Q(s, s^{-1}) \in \mathbf{R}^{p \times m}[s, s^{-1}]$, and $\det P \neq 0$. This comes about because $P(\sigma, \sigma^{-1}): (\mathbf{R}^p)^{\mathbb{Z}} \rightarrow (\mathbf{R}^p)^{\mathbb{Z}}$ is surjective and has a finite dimension kernel.

The I/O dynamical representation will be nonanticipating if and only if $P(s, s^{-1})Q(s, s^{-1}) \in \mathbf{R}^{p \times m}(s)$ is a matrix of proper rational functions. The componentwise partition is accomplished by a matrix Π such that $w = \Pi \begin{bmatrix} u \\ y \end{bmatrix}$.

Recall that the latent variables are accommodated by ARMA representations, and that state variables are a particular type of latent variable representation where the latent variables satisfy the axiom of state. Theorem III tells us that the manifest behavior of the state variable (an ARMA) representation will belong to \mathcal{L}^1 . Consequently, every system $\Sigma \in \mathcal{L}^q$ admits a finite-dimensional state representation. Theorem V specifies that every system $\Sigma \in \mathcal{L}^q$ allows a componentwise I/O representation. Therefore, every system $\Sigma \in \mathcal{L}^q$ admits an input/state/output representation.

Since controllability allows an MA representation and any controllable MA representation can be converted into an AR representation by increasing the lag, complete controllability implies observability. Lack of controllability, however, does not imply lack of observability. This is why inverse modeling or system identification is so difficult. The system and our selection of a representation is critical in that it constrains the behaviors of

the model, affects our ability observe latent variables, and impacts our ability to represent the outcomes U .

As we have seen, latent variables form an important part of representing systems in that they provide a way of formalizing models that contain auxiliary variables. One method of representing latent variables is through state variables. A state-space dynamical system is defined as a dynamical system with latent variables, $\Sigma = (T, W, X, B_s)$ with $X \subseteq L$, such that the full behavior $B_s \in W \times X$ satisfies the axiom of state. In this case the latent variables, the states, contain sufficient information about the past so as to determine future autonomous behavior.

We can combine the above constructs to define a class of models with all of the advantages of completeness, described by the difference equation; state form, the memory is displayed through the latent variables; and nonanticipating input-output, an explicit cause and effect structure. This representation is an input/state/output representation and is the model class most amenable to analysis, synthesis and simulation.

4. IDENTIFIABILITY

Identifiability relates to the ability to reconstruct the dynamical laws of the system from a given set of measurements [4]. The issue is whether the identification procedure will yield a unique value of the parameter, and/or whether the resulting model is the true system. Is the data informative enough to distinguish between different models? If the data are informative enough, will different parameter values give equal models?

4.1. Definitions

First, consider identifiability at a point. A model structure \mathcal{M} is **globally identifiable** at θ_* if:

$$\mathcal{M}(\theta) = \mathcal{M}(\theta_*), \theta \in \mathcal{D}_m \Rightarrow \theta = \theta_*$$

A model structure \mathcal{M} is **strictly globally identifiable** if it is globally identifiable at all $\theta_* \in \mathcal{D}_m$. This is a demanding definition. Strict global identifiability may be lost on hypersurfaces corresponding to lower order systems [5].

A weaker, more realistic definition is global identifiability. A model structure \mathcal{M} is **globally identifiable** if it is globally identifiable at almost all $\theta_* \in \mathcal{D}_m$.

4.2. Discussion

There are several obstructions to identifiability. Feedback makes it difficult to separate system dynamics from the dynamics of feedback. Structured inputs can interfere with the structure of the behavior. Lastly, the failure of the input to excite all of the modes will prevent observation (and subsequent identification) of the unexcited modes.

In order to identify a portion of a system, we must be able to observe the response. Observability specifies the ability to determine the trajectory of latent variables from the manifest set. Since controllability allows an MA representation and any controllable MA representation can be converted into an AR representation by increasing the lag, complete controllability implies observability. Lack of controllability, however, does not imply lack of observability [6]. For systems that can be reduced to an AR representation, the following is a well known result.

Theorem VI. Let $\Sigma = (Z, R^q, R^q, \mathcal{B}) \in \mathcal{L}^{q_1+q_2}$ be represented by an AR-system $R_1(\sigma, \sigma^{-1})w_1 + R_2(\sigma, \sigma^{-1})w_2 = 0$ with $R_1(s, s^{-1}) \in \mathbb{R}^{q_1 \times q_1}[s, s^{-1}]$ & $R_2(s, s^{-1}) \in \mathbb{R}^{q_1 \times q_2}[s, s^{-1}]$. Then w_2 is observable from w_1 in Σ if and only if the rank of the matrix $R_2(\sigma, \sigma^{-1})$ is equal to q_2 for all $\sigma \neq 0$.

This is why inverse modeling or system identification is so difficult. The system and our selection of a representation is critical in that it constrains the behaviors of the model, affects our ability to observe latent variables, and impacts our ability to represent the outcomes U .

The above results guarantee that any unstructured input will be sufficiently rich to observe a controllable system. Structured inputs will allow observation and identification if the AR relations defining the structure of the input have large lags that do not interfere with the structure of the system. In other words, if the structure of the input is not seen by the system.

With respect to metamodeling combat simulations, the systems we are trying to identify are complex, nonlinear, time-varying discrete event systems. In general, for this case, the predictor function is a nonlinear function of past observations and there are too many possibilities for unstructured "black box" models. Knowledge of the nonlinearities must be built into the model. [5]

Fortunately, in this case, we have explicit knowledge of the nature and characteristics of the model. We have the model (the simulation) that applied the system to the inputs to generate the outputs that we are interested in. Given this information, we can build the nonlinearities into the structure of the metamodel and provide the capability to generate a reduced order approximation of the original model. This fact makes metamodeling as a method of model abstraction feasible. We will exploit this fact to the fullest extent possible.

Additional information on Identifiability is provided in Chapter 7 under "Minimal Realizations, Observability, and Identifiability."

5. REPRESENTING DISCRETE EVENT SYSTEMS

The framework in Section II introduces the issues associated with Discrete Event Systems. Most of system identification is formulated on continuous, discrete, or continuous-discrete dynamical systems. Many of the simulations are discrete event or connected discrete-event dynamical systems. The question arises: "When can a DES be described by a difference equation?" Since completeness is usually violated, this impact must be expressly considered. If a linear time-invariant system is not complete, then whether or not $w:Z \rightarrow R^q$ belongs to the behavior depends on $w(t)$ at $t = \pm \infty$. However, results for complete systems can be generalized if the system behavior is restricted to a finite dimensional sequence.

From Theorem II (item 2), every behavior $\mathcal{B} \in \mathcal{L}^q$ allows an AR representation.

Define a DES as a time-invariant system $\Sigma = (Z, W, \mathcal{B})$ with W a finite set. A DES is internally finite if it can be realized by a finite automation, if there exists a state-space representation of it with a finite-state space. A complete DES $\Sigma = (Z, W, \mathcal{B})$ can be described by a behavioral difference equation $f \circ (\sigma^L w, \sigma^{L-1} w, \dots, \sigma^1 w, w) = 0$ for some $L \in Z_+$ and some $f \circ W^{L+1} \rightarrow \{0,1\}$ if the DES is internally finite and if all of its minimal state representations are equivalent.

5.1. Simulation

Mathematical simulation (as opposed to a more general definition of simulation where physical models or environments are used to represent the behavior) is a particular type of model structure that combines the above representations to determine the system behavior. Let $\Sigma = (Z, W, \mathcal{B})$ be a dynamical system. A simulation of Σ is a procedure for selecting an arbitrary element of the behavior \mathcal{B} and defining an algorithm for computing it. An analysis of the construction of parametrization of \mathcal{B} leads to a simulation procedure:

1. Start with $R(\sigma, \sigma^{-1})\omega = 0$.
2. Find a minimal I/S/O representation for this system assuming A is invertible.:

$$\sigma x = Ax + Bu$$

$$y = Cx + Du$$

$$\omega = \Pi \begin{bmatrix} u \\ y \end{bmatrix}$$

3. Choose a vector $x_0 \in R^n$ and a time series and compute ω , via u and y by:

$$x(t+1) = Ax(t) + B\tilde{u}(t) \quad \text{for } t \geq 0$$

$$x(t-1) = A^{-1}x(t) - A^{-1}B\tilde{u}(t-1) \quad \text{for } t \leq 0$$

with

$$x(0) = x_0$$

$$\text{and } y(t) = Cx(t) + Du(t) \quad \text{for } t \in Z$$

5.2. Discrete Event Nature of Modeled Events

Military engagement simulations usually are defined to represent real-world events that have a beginning and an end. Given that the data includes the behavior that is to be modeled and that the simulation terminates naturally, results for complete systems can be applied since the system behavior is restricted to a finite dimensional sequence and the axiom of state is assumed in the definition of initial conditions.

In general, the axiom of state is assumed because the simulation is set up in such a way that the initial conditions contain sufficient information about the past so as to determine future autonomous behavior. Also, an input-output structure with causality is assumed and evident in the presence of input and output files.

This only leaves the question of the information available in the data. Is the data sufficient? The question arises: "What are unnatural terminations and when can the results of finite dimensional complete systems be applied? The answer comes from the definition of a dynamical system and must consider the behavior that is to be modeled, the representation selected/desired, and the data. For a stochastic system with multiple realizations, the ensemble of trajectories must span the space. Any single trajectory, for a stochastic or deterministic system, must span both the input and output space and be sufficiently long so that the state transition probabilities also span the allowable probability space and the distribution of these probabilities are the same as the underlying system. This condition can be assumed if the simulation reaches equilibrium. In this case, additional run time does not change the state of the simulation.

If the simulation does not reach equilibrium, there may still be adequate information in the data. This condition, however, cannot be verified without further testing.

In summary, assuming that the underlying system modeled by the simulation is well behaved (Markovian, complete with respect to the modeled behavior), the following is required to metamodel combat simulations:

1. The data must include the behavior we are trying to model.
2. The latent variables that define the behavior must be observable.
3. The input must be persistently exciting so that the effects of the latent variables are observed.
4. For a stochastic system, the ensemble of trajectories must span the space.
5. Any single trajectory must span both the input and output space and be sufficiently long so that the state transition probabilities also span the allowable probability space and the distribution of these probabilities are the same as the underlying system.

6. REFERENCES

1. J. C. Willems, "Paradigms and Puzzles in the Theory of Dynamical Systems," *IEEE Trans. on Automat. Contr.*, vol 36, no. 3, pp 259-294, March 1991.
2. A. W. Naylor, and G. R. Sell. *Linear Operator Theory in Engineering and Science*, New York: Springer Verlag, 1982.
3. H. L. Royden, *Real Analysis*, Macmillan Publishing Company, 1988.
4. T. Kailath, *Linear Systems*, Prentice-Hall, New Jersey, 1980.
5. L. Ljung, *System Identification: Theory for the User*, Prentice-Hall, New Jersey, 1987.
6. H. Kwakernaak and R. Sivan, *Linear Optimal Control Systems*, Wiley-Interscience, 1972.

CHAPTER 4

AIR FORCE METAMODELING PROBLEMS

1. CHAPTER OUTLINE

1. Chapter Outline	4-1
2. Introduction.....	4-3
2.1. Chapter Summary	4-3
2.2. Definitions	4-3
2.3. Research Focus	4-3
2.4. Research Outline.....	4-3
3. Metamodel Requirements (Purpose of the metamodel).....	4-4
3.1. Introduction.....	4-4
3.2. Overview of Metamodel Purpose	4-4
3.3. Acquisition Metamodels.....	4-4
3.3.1. Objective.....	4-5
3.3.1.1. Analytical Metamodels.....	4-5
3.3.1.2. Simulation Metamodels.....	4-6
3.4. Operational Metamodels	4-8
3.4.1. Objective.....	4-8
3.4.1.1. Planning.....	4-8
3.4.1.2. Training	4-8
3.4.1.3. Modify the Concept of Operations.....	4-8
3.5. Summary of Metamodel Purpose.....	4-9
4. Simulation Characteristics.....	4-10
4.1. External Interface - SIMTAX.....	4-10
4.1.1. Purpose.....	4-10
4.1.1.1. Analysis	4-10
4.1.1.2. Training and Education	4-11
4.1.2. Qualities.....	4-11
4.1.2.1. Domain.....	4-12
4.1.2.2. Span.....	4-12
4.1.2.3. Environment	4-13
4.1.2.4. Force Composition.....	4-14
4.1.2.5. Scope of Conflict	4-14
4.1.2.6. Mission Area.....	4-15
4.1.2.7. Level of Detail of Processes and Entities	4-15

4.1.3.	Construction.....	4-15
4.1.3.1.	Human Participation.....	4-15
4.1.4.	Time Processing	4-16
4.1.5.	Treatment of Randomness	4-16
4.1.6.	Sidedness	4-17
4.1.7.	Additional Catalog Data	4-18
4.2.	Internal Processing.....	4-18
4.2.1.	Basis	4-19
4.2.2.	Process Description	4-19
4.2.3.	System, Input, and Output Processing.....	4-19
4.2.3.	Remaining Internal Process Selections	4-20
4.3.	Summary of Simulation Characteristics	4-21
5.	Wargaming and Simulation matrix	4-22
5.1.	Scope.....	4-22
6.	Metamodeling classes	4-24
6.1.	Metamodeling Feature Space	4-24
6.2.	Metric Spaces	4-24
6.3.	Nonparametric Clustering Techniques	4-24
6.4.	Binary Vector Space Analysis	4-25
6.5.	Results.....	4-27
7.	References	4-28

2. INTRODUCTION

2.1. Chapter Summary

This chapter defines the classes of Air Force metamodeling problems based on the simulations and *a priori* knowledge (metamodel use). It defines these classes by analyzing the requirements and simulation characteristics, determining criteria for clustering metamodeling problems, and then applying these criteria to selected simulations.

2.2. Definitions

We have defined a metamodeling problem as the direct sum of the model (simulation) and metamodel requirements. This means that the same simulation could be part of two different metamodeling problems if the requirements were different. Or conversely, the same set of requirements applied to two different (nonsimilar) simulations also leads to two different metamodeling problems. Therefore, in order to categorize metamodeling problems, each of these aspects was analyzed independently.

2.3. Research Focus

This chapter focuses on the steps that provide the prior knowledge (the first eight steps):

1. Determine the purpose of the metamodel
2. Identify the response
3. Identify important response characteristics
4. Identify input factors
5. Identify important input characteristics
6. Specify the experimental region
7. Select validity measures
8. Specify required validity

Step 1, the purpose of the metamodel (analysis or hierarchical simulation) is independent of the simulation that will be metamodeled. All of the remaining steps are a function (direct sum) of both the metamodel requirements and the simulation that is to be modeled. Consequently, it is not possible to build the classes of Air Force metamodeling problems by independently considering the range of solutions to each of these steps. Therefore, the research concentrated on the aggregate space of metamodel requirements and simulation characteristics and did not specifically address each step.

2.4. Research Outline

The connection between prior knowledge and the metamodeling technique began with an analysis of the types of problems facing the Air Force analyst and engineer. In order to group these problems for use with an identification technique, significant characteristics of these problems were identified. These characteristics defined classes of Air Force metamodeling problems based on the simulations and *a priori* knowledge (metamodel use).

Once the feature space that encompasses the selected metamodeling problems was defined, the next step was to determine classes of metamodeling problems. This was accomplished by evaluating the density of the metamodeling problems in the feature space and, based on the characteristics of the space, selecting criteria for clustering. Classes of metamodeling problems were then defined by these clusters.

3. METAMODEL REQUIREMENTS (PURPOSE OF THE METAMODEL)

3.1. Introduction

In this section, the discussion on simulation in the Air Force (Appendix I) is used to define the purpose of the metamodel. Recall that Step 1, the purpose of the metamodel is independent of the simulation that will be metamodeled.

3.2. Overview of Metamodel Purpose

There are two independent constructs for the analyses of metamodel requirements. The first is the type of Air Force program (area) that the metamodel is used to support, the second is the specific objective of the metamodel.

From the discussion above, there are two general types of programs that metamodels can support. These are:

1. Acquisition (including the total integrated weapon system support - Phase IV) and,
2. Operations, including the deployment (logistics) as well as the employment of the system.

In addition to, and independent of, the program that the metamodel will support there are two general objectives of metamodels. As mathematical relationships, metamodels can be developed to support two general purposes:

1. Analysis
2. Hierarchical simulation

First, a metamodel can be used for analysis. In this case, the metamodel becomes an independent structure that is used to understand and extract information from the model.

Secondly, a metamodel can be used to support hierarchical simulation and model reuse. In this case, the metamodel is used in conjunction with (coupled to) other simulations or simulation elements to answer larger questions that are not supported within the structure of the modeled simulation.

3.3. Acquisition Metamodels

Acquisition metamodels are based on simulations that are used to: (1) develop the system concept; (2) mature the design of the system; and (3) define the concept of operations. They can be used in any of the phases (Phase 0 through IV) and at any level of analysis.

3.3.1. Objective

Acquisition metamodels are defined for both objectives: analysis and hierarchical simulation.

3.3.1.1. Analytical Metamodels.

There is a tradeoff between model complexity and our ability to perform analysis on the system via the model [1]. The higher the level of abstraction, the simpler the analysis. In the analysis of complex systems, the total space of interest is first considered at a lower resolution. Only a subset of interest is usually considered at the higher resolution. This process continues until the highest (desired) resolution is achieved. This consecutive focusing of attention results in a multilevel task decomposition [2].

Analytical metamodels can be developed to approximate an unknown response surface (determine relationships) or to model a simulated process. Once developed, analytical metamodels can be used to understand relationships, optimize expected performance, or predict future responses.

3.3.1.1.1. Scope

3.3.1.1.1.1. *Approximate an Unknown Response Surface*

Response surface methodology, or RSM, is a collection of mathematical and statistical techniques that are useful for modeling and analysis of problems in which the response of interest is influenced by several variables [3]. The "response surface" is the graph of the expected response as a function of the input variables.

Approximation of the response surface by a metamodel is the first step in most RSM problems. Use of metamodels to approximate a response surface, however, requires a transfer function or an explicit Input-Output map. Once developed, the metamodel provides a general understanding of the modeled process and can support additional analysis outlined below (sensitivity analysis, estimation of existing states, or optimization).

3.3.1.1.1.2. *Model a Simulated Process*

A process is a function, operator, algorithm, or procedure that operates on some domain and produces a result. Modeling a simulated process is different from the approximation of an unknown response surface because it does not require the development of a complete Input-Output multivariable map. In fact, the modeling of a simulated process may be an intermediate step in a high fidelity approximation of the Input-Output map.

In this case, inverse modeling (developing a model from the data) is used to identify the process without regard to Input-Output transformation. The inputs and outputs of the metamodel may actually be latent variables that are not observed in the simulation input or output.

3.3.1.1.2. Uses

To make any sense, identification of a metamodel must have some use [4].

3.3.1.1.2.1. Sensitivity Analysis

Models used for this purpose extract the essentials from complicated evidence and quantify the implications. Once the coefficients of the mathematical model are assumed known, they can be held constant. Now the input data can be changed to see the effect of these changes on the output.

3.3.1.1.2.2. Estimation of Existing States

The object of state estimation is to track variables, which may characterize some dynamical behavior, by processing observations that include errors. State estimators rely on a model. Again, the coefficients of the mathematical model are assumed known through the identification of a metamodel. New observations are taken and an estimate of the state can be made.

3.3.1.1.2.3. Predict and Control Future Responses

Prediction is important if the metamodel is used as a model to control future events. Use of a metamodel to predict future responses is a complicated issue (may not be appropriate) and is a function of the domain and range of the metamodel.

3.3.1.1.2.4. Optimize Expected Performance

In this case, we are not using a metamodel to gain an understanding of the process mechanism. Instead, we are trying to find the combination of input variables that maximize the response subject to external constraints. This is the classical use of RSM and requires the ability to control input variables.

3.3.1.1.2.5. Diagnosis of Faults

A great benefit of metamodeling is the ability to uncover anomalies and shortcomings. In complex simulations, it is not possible to directly correlate a single input with the outputs. In the development of a metamodel, the influence of an input variable may be significantly more or less than expected. If subject matter experts (experience) directly conflicts with this finding, an error in the original simulation may be the cause.

3.3.1.2. Simulation Metamodels

The other purpose of a metamodel is to support simulation. This support is based on the hierarchical representation discussed in Appendix II. Using metamodels for this purpose is a two-step process. First a metamodel of a simulation (or component) is generated to develop more abstract simulation models. Then, once developed, these modules can be

used to couple these metamodels (modules) to other simulations or metamodels to simulate a more complex system.

3.3.1.2.1. Scope

3.3.1.2.1.1. Develop Atomic Simulation Components

This is the process of actually metamodeling a simulation, or a component of the simulation.

3.3.1.2.1.2. Build Coupled Simulation Components

Given two or more simulation components (metamodels), these models can be coupled to provide a more complex simulation or simulation components. If a coupled component does not complete the simulation requirements, then this coupled component is coupled back into the simulation replacing the components that were metamodeled.

3.3.1.2.2. Uses

3.3.1.2.2.1. Execution Speed

Since a metamodel, as we have defined it, is a straightforward mathematical relationship, it can execute much faster than the simulation component that it replaced. As a result, the overall simulation will execute faster.

3.3.1.2.2.2. Maintainability / Configuration Control

Often, large simulations can become unmanageable and difficult to maintain. The use of a metamodel in place of some simulation components will accomplish several things. First, it will encourage more thorough testing. Since the execution of the overall simulation will be quicker, changes can be tested in a more reasonable time. Second, development of the metamodel will identify input elements that may have been included in the code but are not actually required by the simulation component.

3.3.1.2.2.3. Verification, Validation, and Accreditation

Verification and validation of simulations is a complex task. Current procedures are manpower intensive and still subject to interpretation (see [5] for example). Metamodeling provides a feasible alternative. Assume that we have an accredited simulation and a slightly modified version that is supposed to provide essentially the same overall result with additional data for some element of the simulation. By subjecting the two different simulations to the same environment, developing a metamodel for each simulation from the data, the parameters of the two metamodels can be directly compared. The differences are direct.

3.4. Operational Metamodels

Because the operational community participates in the acquisition process and has the responsibility to define requirements, the operational community may use exactly the same simulations as the acquisition community. In addition, they may use them for exactly the same purpose. Requirements for these metamodels are considered under the previous section (acquisition metamodels).

In addition to the acquisition process, the operational community also uses M&S in support of planning, exercises, and real operations. These are the metamodel requirements considered next. Use of these models and simulations will be discussed contrasted with acquisition M&S we have already discussed.

3.4.1. Objective

3.4.1.1. Planning

M&S used during for operational planning are similar to analytical metamodels used for acquisition. Since these models and simulations are primarily used to determine existing relationships or to optimize plans, their application is usually very specific to a given scenario. With two major exceptions, operational models and simulations are not distinguishable from those used during the acquisition phase. The two exceptions follow.

First, operational M&S does not generally require high fidelity engineering level models and simulations capable of representing the internal details of a weapon system. Second, some of the operational planning systems may be trying to capture the interactions present in a large worldwide network. Consequently, these M&S applications may be better classified as representing systems of systems.

3.4.1.2. Training

While M&S applications from acquisition could be applied to training systems, the requirements for training systems are quite different. Since real time execution is often required, these systems are usually low fidelity approximations that emphasize the "look and feel" of the simulated system.

3.4.1.3. Modify the Concept of Operations

M&S used to support formal (high level) modifications to concepts of operations are usually the same combat engagement models and simulations used for the acquisition phase. Beyond the headquarters level, the concept of operations for a particular unit or system is dependent on both the system and the environment. Since the environment is so complex and transient, models used to modify operational concepts are usually straightforward and subjective.

3.5. Summary of Metamodel Purpose

Table 4.3.1 summarizes the use of metamodels in the acquisition process, while Table 4.3.2 summarizes the purpose of metamodels.

Table 4.3.1. Metamodel Use in the Acquisition Process.

PROGRAM	METAMODEL USE
Phase	Mission Area Analysis Concept Exploitation and Definition Demonstration and Validation Engineering and Manufacturing Development Production and Deployment Operations and Support
Level	1. Engineering Analysis (Level I) 2. Weapon System Capability (Level II) 3. Combat Capability (Level III) 4. Campaign Results (Level IV)
Objective	Analytical Hierarchical Simulation

Table 4.3.2. Metamodel Purpose Summary.

OBJECTIVE	SCOPE	USES
Analytical	Approximate an unknown response surface Model a simulated process	Sensitivity analysis Estimation of existing states Predict and control future responses Optimize expected performance Diagnosis of faults
Simulation	Develop atomic simulation components Build coupled simulation components	Execution speed Maintainability/Configuration control Verification, validation, accreditation

4. SIMULATION CHARACTERISTICS

We have discussed Step 1, the purpose of the metamodel. Since all of the remaining steps are a function (direct sum) of both the metamodel requirements and the simulation that is to be modeled, we now concentrate on the aggregate space of simulation characteristics.

We suggest a space that consists of a general description of the simulation or model as well as further detail on the process structure of the internal components. The general description follows "SIMTAX, A Taxonomy for Warfare Simulation" developed by the Military Operational Research Society (MORS) [6]. This is a descriptive framework designed to guide the development, acquisition, and use of warfare models and provides the basis for classifying objects for identification, retrieval, and research purposes.

While the general description provides a basis for the taxonomy, prior research indicated that it was not sufficient to categorize metamodeling problems with enough clarity to define a connection between prior knowledge and a metamodeling technique [7,8]. To support this connection and provide a link between the more general taxonomy and the metamodeling technique, a more detailed internal taxonomy was appended to the SIMTAX. This additional detail is classified under "internal processing."

4.1. External Interface - SIMTAX

The MORS workshop concluded that a taxonomy for warfare simulations should address three equally important relational dimensions: the purpose, the qualities, and the construction of the model or simulation.

4.1.1. Purpose

The purpose explains why the model was built or to what use the model could be applied. There are two major divisions: analysis, and training and education. In addition to a stated purpose, this can be addressed by the **Model type**:

4.1.1.1. Analysis

Simulations developed to discover, deduce, or expand relationships or lessons learned are analytical simulations.

Table 4.4.1 lists allowable selections for analytical simulations.

Table 4.4.1 Selections for Analytical Simulations.

ABBREVIATION	DESCRIPTION
A	Analysis
A-OST	Analysis, operation support tool (decision aid)
A-RE	Analysis, research and evaluation tool
A-RE-CD	Analysis, research and evaluation tool dealing with combat development
A-RE-FCR	Analysis, research and evaluation tool dealing with force capability and requirements
A-RE-WS	Analysis, research and evaluation tool dealing with weapon systems

4.1.1.2. Training and Education

Training and education simulations are designed to transfer or reinforce information to improve the proficiency in the conduct of war. Table 4.4.2 lists allowable selections for training and education simulations.

Table 4.4.2. Selections for Training and Education Simulations.

ABBREVIATION	DESCRIPTION
EDU	Education
T/E	Training and education
T/E-ED	Training and education, exercise driver
T/E-SD	Training and education, skills development
TR	Training

4.1.2. Qualities

The qualities dimensions are those entities and processes which the model represents. This dimension is often covered as the **Description**:

4.1.2.1. Domain

This is the physical or abstract space in which the entities and processes operate. Table 4.4.3 lists allowable selections for the domain.

Table 4.4.3. Selections for the Domain.

ABBREVIATION	DESCRIPTION
A	Air
AB	Airbase
ABS	Abstract
CO	Coast
L	Land
N	Naval
POL	Politics
S	Sea
SP	Space
US	Undersea

4.1.2.2. Span

The span is the scale of the domain. Table 4.4.4 lists allowable selections for the span.

Table 4.4.4. Selections for the Span.

ABBREVIATION	DESCRIPTION
GEO	Geographic area
GLOB	Global
IND	Individual
INTER	Intertheater
INTRA	Intratheater
LOC	Local
REG	Regional
SECT	Sector
TH	Theater

4.1.2.3. Environment

The environment is the detail of the domain (see Table 4.4.5).

Table 4.4.5. Environment.

ABBREVIATION	DESCRIPTION
A	Air
BAR	Barrier
BAT	Battlefield
CAN	Canalization
CF	Cultural features
COM	Communications
DES	Deserts
D/N	Day and night
DT	Digitized terrain
EAR	Earth
EW	Electronic warfare
FOR	Forestation
GEO	Geography
HEX	Hex-based
JU	Jungles
L	Land
MET	Meteorological conditions
S	Sea
SEAS	Seasons
SP	Space
SS	Sea states
S/S	Sunrise and sunset
TD	Time of day
TEMP	Temperature
TER	Terrain
TF	Transportation factors
TRAF	Trafficability
URB	Urban
UW	Underwater
VEG	Vegetation
W	Weather

4.1.2.4. Force Composition

This descriptor is the mix of forces that can be portrayed by the model. Table 4.4.6 lists allowable selections for force composition.

Table 4.4.6. Selections for Force Composition.

ABBREVIATION	DESCRIPTION
AB	Airbase
COMB	Combined
CONC	Conceptual
COMP	Component
CORPS	Corps
ELEM	Element
JF	Joint

4.1.2.5. Scope of Conflict

This is the category of weapons or systems simulated (see Table 4.4.7).

Table 4.4.7. Category of Weapons or Systems Simulated.

ABBREVIATION	DESCRIPTION
BIO	Biological
CH	Chemical
CONV	Conventional
DET	Detection
ELEC	Electronic combat/warfare
KIN	Kinetic
LAS	Laser
MIN	Mines
NONSTR	Nonstrategic
NUC	Nuclear
POL	Political
RA	Rear area
SPEC	Special
STRAT	Strategic
UNC	Unconventional
VER	Verification

4.1.2.6. Mission Area

This is the recognized combination of weapons and procedures used to accomplish a specific objective. Table 4.4.8 lists allowable selections for mission area.

Table 4.4.8. Selections for Mission Area.

ABBREVIATION	DESCRIPTION
CAS	Close air support
AL	Airlift
SC	Sea control
IA	Indirect artillery

4.1.2.7. Level of Detail of Processes and Entities

This category of the quality dimension has two components: entities and processes. The level of detail describes the lowest discrete entity modeled. There are five general methods. Weapons systems can be accounted for individually, by number of types of systems, by the number of weapons systems, by the groups of weapons systems by unit, or finally, by groups of weapons systems by generic types of units.

The process affects the entities. Attrition, communications, and movements are examples of process. Each of these processes can have its own taxonomy. For example attrition can be accounted for by Monte Carlo techniques, by homogeneous or heterogeneous Lanchester square (difference or differential) equations, by homogeneous or heterogeneous Lanchester linear (difference or differential) equations, etc.

4.1.3. Construction

The construction defines the design of the model. There are four major categories: human participation, time processing, treatment of randomness, and sidedness.

4.1.3.1. Human Participation

This category defines the extent to which human presence is allowed or required to influence the operation of the model. There are two major branches: required, and not required. Table 4.4.9 lists allowable selections for human participation.

Table 4.4.9. Human Participation.

ABBREVIATION	DESCRIPTION
NP	Not permitted
NR	Not required
NR-INT	Not required, model interruptable
NR-SC	Not required, model has scheduled changes
REQ	Required
REQ-A	Required for analysis
REQ-D	Required for decisions
REQ-P	Required for processes
REQ-DP	Required for decisions and processes
REQ-GR	Required for graphics
REQ-I	Required for input
REQ-ID	Required for interactive decisions
REQ-SU	Required for setup
U-I	User-interactive

4.1.4. Time Processing

There are also two major divisions in this category. These are "dynamic" models that represent time dependent process, and "static" models that do not have a dependence on time. "Dynamic" processes are further divided depending on the manner that time passes. Time can run continuously, or it can step through time. These steps can be a fixed increment of time (time step model) or it can step through time as a function of a set of events that must be accomplished (event step model). Table 4.4.10 lists allowable selections for time processing.

Table 4.4.10. Selections for Time Processing.

ABBREVIATION	DESCRIPTION
DYN	Dynamic
DYN-CF	Dynamic, closed form
DYN-ES	Dynamic, event-step
DYN-TS	Dynamic, time-step
STATIC	Static
STATIS	Statistical

4.1.5. Treatment of Randomness

Stochastic models acknowledge the possibility of various outcomes and are either direct computation or Monte Carlo (if any part of a stochastic simulation is Monte Carlo it is classified as a Monte Carlo simulation). For each run (trial), a Monte Carlo simulation precludes one realization of the process by "drawing" (at least once) pseudorandom

numbers to determine an outcome. These trials can be combined to determine the expected value of the outcome.

Deterministic models do not represent variations in outcomes. This branch (deterministic models), however, may still generate a value as a function of an expected value. In this case we have a deterministic model of a stochastic process. (We have replaced the random variable with its expected value - it is still a deterministic model). Table 4.4.11 lists allowable selections for the treatment of randomness.

Table 4.4.11. Treatment of Randomness.

ABBREVIATION	DESCRIPTION
DET	Deterministic
DET-EV	Deterministic, generates value as a function of an expected value
STO	Stochastic
STO-DC	Stochastic, direct computation
STO-MC	Stochastic, Monte Carlo

4.1.6. Sidedness

Sidedness refers to the number of collections of resources working toward a common goal. Simulations are classified as one, two or three or more sided. Two-sided and three-sided models are symmetric or asymmetric. A symmetric model allows either side to use a particular set of weapons systems and/or tactics. An asymmetric model places restrictions on this use, and one or both sides can be reactive or non-reactive to the actions of the other side. The number of resources in one-sided models is: (a) one, (b) few, 2 through 6; or (c) many, 7 or more. The number of resources in asymmetric or three or more sided models uses the same designation as above and is the maximum number of resources simulated on any given side. Table 4.4.12 lists allowable selections for sidedness.

Table 4.4.12. Sidedness.

ABBREVIATION	DESCRIPTION
1	One-sided
1NR	One side nonreactive (same for reactive)
2	Two-sided
3	Three-sided
A	Asymmetric
NR	Nonreactive
R	Reactive
RED-NR	RED side nonreactive (same for BLUE side)
S	Symmetric

4.1.7. Additional Catalog Data

Although not a part of SIMTAX as defined by the workshop report, the **Input** and **Output** is usually found in most catalogues:

4.2. Internal Processing

Selection of a metamodel structure will require detailed information not contained in the simulation and model catalogues. To provide a link between the more general taxonomy outlined above and specific metamodeling techniques, a more detailed internal taxonomy was appended to the SIMTAX. The purpose of this additional detail is to describe the structure of the simulation in terms of system theoretic definitions common to control engineering. Figure 4.4.1 depicts the model of a continuous system with a sampled measurement. In development of a metamodel, we try to isolate and identify each of the individual elements in this model. Consequently, we must be able to characterize the type of processing that takes place in each of the blocks.

To explain this concept, we will consider the model of an aircraft. The input, $u(t)$, is the pilot or autopilot command. In modern aircraft this would be a desired acceleration ("g") or angle of attack (" α "). In older aircraft it would be something closer to the flight control surface such as the torque necessary to hold the control surface in a given position. This input is acted on by $B(t)$ to provide the input expected by the plant. In an inertial frame it could be the force applied. In a more complicated simulation it could be the control surface deflection. Another input path accepts input disturbances $w(t)$. The plant, represented by $F(t)$, is the model of the physical system. In the case of the aircraft, it could be something as simple as $F = ma$ (if the simulation was completely in an inertial frame) or it could be the body axis stability derivatives that make up the coefficients in the equations of motion. The output, $z(t_i)$ is the measurement available. The instrumentation system that performs these measurements is represented by $H(t)$. The aircraft would have accelerometers or an inertial system that measures the body axis accelerations or inertial position and attitude. The combination of all of these blocks represents a single process or entity in a simulation.

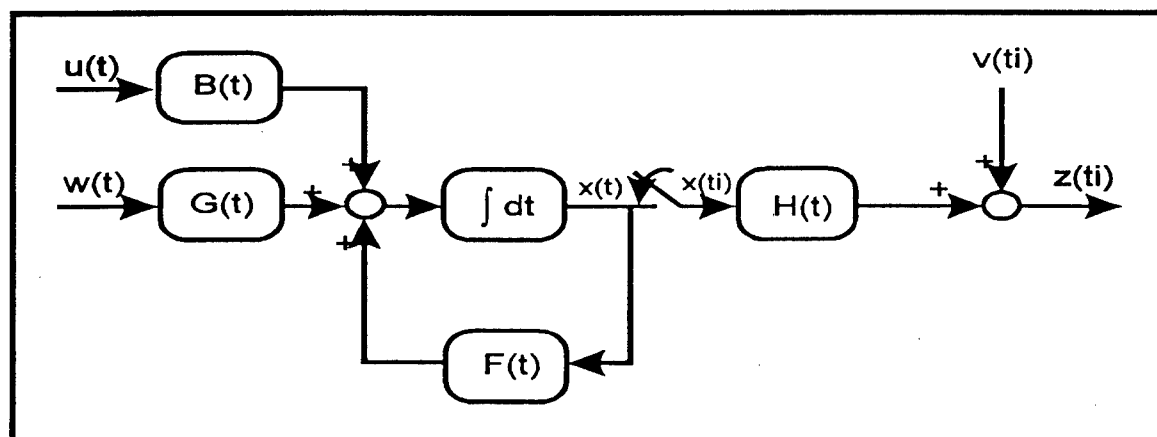


Figure 4.4.1. Sampled-Data Continuous System Model.

From the above discussion we see that we are going to analyze a simulation and tear it into its components that represent elements similar to Figure 4.4.1. Therefore, while the SIMTAX defines simulations as deterministic or stochastic, the internal processing information will isolate the stochastic elements to specific part of the model -- $B(t)$, $F(t)$, or $H(t)$.

4.2.1. Basis

This is the fundamental basis of the simulation. The simulation will model either a physical phenomenon or will model events that simulate human or system interactions with its environment. Simulations that are a combination of the two will default to event based (the more complex of the two basis). Table 4.4.13 lists definitions for the basis.

Table 4.4.13. Definition of Basis for Internal Processing.

SELECTION	DESCRIPTION
Physics based	Output is a function of a physical law
Event based	Output is based on intelligent processing not limited to physical laws

4.2.2. Process Description

Table 4.4.14 defines the process description. This is a description of the entire simulation (component) that will be used to develop the metamodel. The component could be a single routine or function, or an entire simulation.

Table 4.4.14. Definition of Process Description for Internal Processing.

TYPE	DESCRIPTION	VALUE
Complex	Inputs to more than 1 separable process (system)	Number of systems that allow input
Simple	Inputs to only 1 process (system) No additional influence on the system (other than predefined parameters)	Order of the model
Coupled	Inputs to only 1 process (system) Additional non-deterministic impacts on the output	Order of the model

4.2.3. System, Input, and Output Processing

This is the plant or system that is modeled, the techniques used to process the inputs and the method of generating the observed output. Each of these elements is considered independently. Table 4.4.15 lists options for these components.

Table 4.4.15. Options for System, Input, and Output Processing.

SELECTION	DESCRIPTION	VALUE
Algebraic structure	Linear	
	Nonlinear	
Realizations	Stochastic: variables are functions of random variables (includes disturbances)	
	Deterministic: variables are not functions of random variables (does not include disturbances)	

4.2.3. Remaining Internal Process Selections

The result or trajectory of the simulation must be considered. The standard result is the simulated system response. Occasionally, however, simulation results are not complete representations, but are designed to provide data used to build statistical databases. The final two selections pertain to the overall simulation and provide a description of the level of the system and how the trajectory propagates in time. Table 4.4.16 lists options for these components.

Table 4.4.16. Options for Selection of the Internal Process.

SELECTION	DESCRIPTION	VALUE
Result/Trajectory	Functional	
	Statistical Base	
Level:	SISO: Single Input Single Output	Number of inputs - Number of outputs
	MISO: Multiple Input Single Output	
	MIMO: Multiple Input Multiple Output	
Interval	Continuous time	
	Discrete time	
	Continuous - Discrete time	
	Continuous system - discrete (sampled data) measurements	
	Discrete-event	

4.3. Summary of Simulation Characteristics

The following table summarizes the external and internal elements of the simulation feature space.

Table 4.4.17. Metamodel Purpose Summary.

PURPOSE	DESCRIPTION	VALUE
Model type		
Description	Domain Span Environment Force composition Scope of conflict Mission area Level of detail of processes and entities	
Construction	Human participation Time processing Treatment of randomness Sidedness	
Input		
Output		
Internal processing	Basis	Physics based Event based
	Process description	Complex Simple Coupled
	System	Linear Nonlinear Stochastic Deterministic
	Input	Linear Nonlinear Stochastic Deterministic
	Output	Linear Nonlinear Stochastic Deterministic
	Result / Trajectory	Functional Statistical base
	Level	SISO MISO MIMO
	Interval	Continuous time Discrete time Continuous - discrete time Continuous system - discrete (sampled data) measurements Discrete-event
General Data	Data base CPU time per cycle Data output analysis	

5.1. Scope

Table 4.5.1. Wargaming and Simulation Matrix.

4-22

Table 4.5.1. Wargaming and Simulation Matrix (cont.).

CATEGORIES	TYPES	SELECTED
UNCONVENTIONAL WARFARE		
CRISIS ACTION SIMULATIONS		
FORCE ACCOUNTING/ FORCE STRUCTURE		
COMMAND, CONTROL, COMMUNICATIONS, AND INTELLIGENCE (less strategic systems)		X
ELECTRONIC WARFARE		X
INTELLIGENCE		X
WEAPONS SYSTEMS SIMULATIONS	Air Systems Rotary Wing Air Systems Fixed Wing Ground Systems Air Defense Special Systems Chemical Systems Weapon Systems, Generic	X X X
LOGISTICS		
MOBILIZATION AND INDUSTRIAL PREPAREDNESS		
TRANSPORTATION AND MOBILITY		
MEDICAL		
ECONOMIC		
ENVIRONMENTAL EFFECT		
MISCELLANEOUS		
SPACE		
WEATHER		
LOW INTENSITY CONFLICT		

6. METAMODELING CLASSES

6.1. Metamodeling Feature Space

This research has demonstrated that the most important issue with respect to metamodeling is the type and number of processes that are modeled by the simulation. The analysis considered here, however, was accomplished without having the source code for the simulation and only considers external factors (the SIMTAX feature space) to categorize the simulations. Using the simulations from Table 4.6.1, a binary feature space of dimension 125 was developed.

In order to evaluate the density of the metamodeling problems in this feature space, a metric is required that describes the closeness (distance) of one simulation to another. Selection of the proper metric is critical to that analysis [10].

6.2. Metric Spaces

A metric space is a pair of objects, a set X and a metric, or distance function, d . The metric $d(x,y)$ is a real valued function satisfying the following axioms [11]:

$$\begin{aligned}d(x,y) &> 0 \quad d(x,x) = 0 \quad \forall x,y \in X \\ \text{if } d(x,y) &= 0 \implies x = y \quad \forall x,y \in X \\ d(x,y) &= d(y,x) \quad \forall x,y \in X \\ d(x,y) &\leq d(x,z) + d(z,y) \quad \forall x,y \in X\end{aligned}$$

For the type of information we are trying to extract from the feature vector, we can define the following metric:

Let $X(n)$ is the set of all ordered n -tuples of "zeros" and "ones." For example:

$$X(2) = \{00, 01, 10, 11\}$$

Let $d(x,y)$ = the number of places where x and y have different entries. Then $(X(n), d)$ is a metric space that satisfies the above axioms.

6.3. Nonparametric Clustering Techniques

If the clusters are separated, even if they are a different diameter, **separating hyperplanes** (decision surfaces) can be defined that associate a measurement with a cluster. This technique will not work with overlapping clusters. The **K-nearest neighbor** technique should be used for overlapping clusters that are equally likely. In this case, the number of points in the cluster define *a-priori* probability of occurrence.

The **nearest prototype** technique will provide the nearest centroid regardless of the dispersion. Each class/aspect combination is represented by a cluster centroid, and each element is assigned to the cluster with the shortest measurement-to-prototype distance.

Classification by any of these clustering techniques will not provide an indication of the confidence of the choice.

6.4. Binary Vector Space Analysis

An analysis of the total binary vector space (cardinality = 162, dimension = 125) where D_{ij} is the difference between feature vectors showed that:

$$\min_{i,j \in N} (D_{ij}) = 1 \quad i \neq j$$

and, therefore, no two simulations are the same and $(X(n), d)$ is a valid metric space (computed by $\min(Y(2,:))$ in sortbyd.m). Also:

1. The maximum distance between any two simulations $\left(\max_{i,j \in N} (D_{ij}) \right)$ was 34.
2. The maximum value $\left(\max_{j \in N} \left(\min_{i \in N} (D_{ij}) \right) \right)$ of the minimum separation between any two simulations was 20..
3. The average closest distance between all simulations is 7.44.
4. The standard deviation of the closest distance is 3.64.

Further use of difference between feature vectors (D_{ij}) to develop clusters of simulations is being investigated. Since this measure primarily provides a direct indication of the complexity of the simulation, clusters based on the distance may provide useful groups for defining metamodel structures.

A characteristic vector was defined for each subcategory entered into the database. This vector was used as the centroid of a cluster for the category and the distance from each simulation to all of the characteristic vectors (one defined for each category) was determined and sorted by distance for the center of the category of which it was a member. Figure 4.6.1 is a plot of the distance of the simulations from its cluster center.

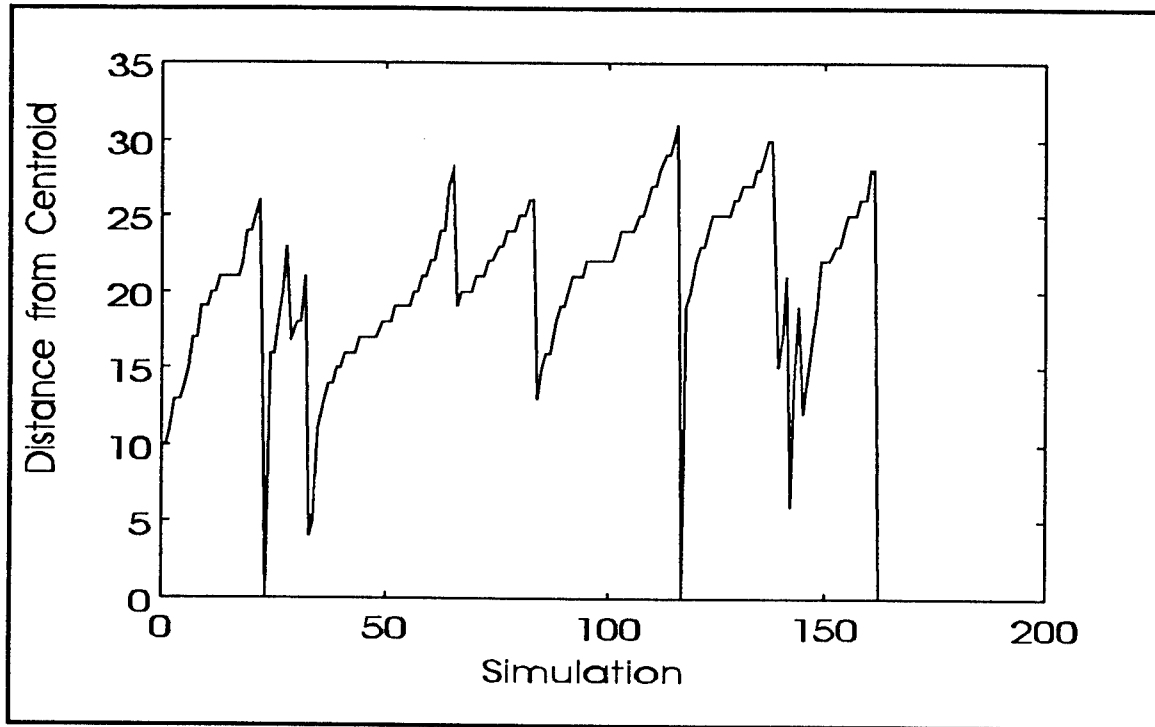


Figure 4.6.2. *Distance of Each Simulation from the Center of its Category Center.*

From this plot (and the data) we see that there are 3 clusters with a single simulation and that there are six additional clusters with less than 5 simulations. These eight clusters should eventually be consolidated with the seven larger clusters.

Figure 4.6.3 is a plot of the distance of all of the simulations from the centroid of the first category. This plot shows that many of the simulations which are not in the cluster are within its boundary and that the clusters require further refinement.

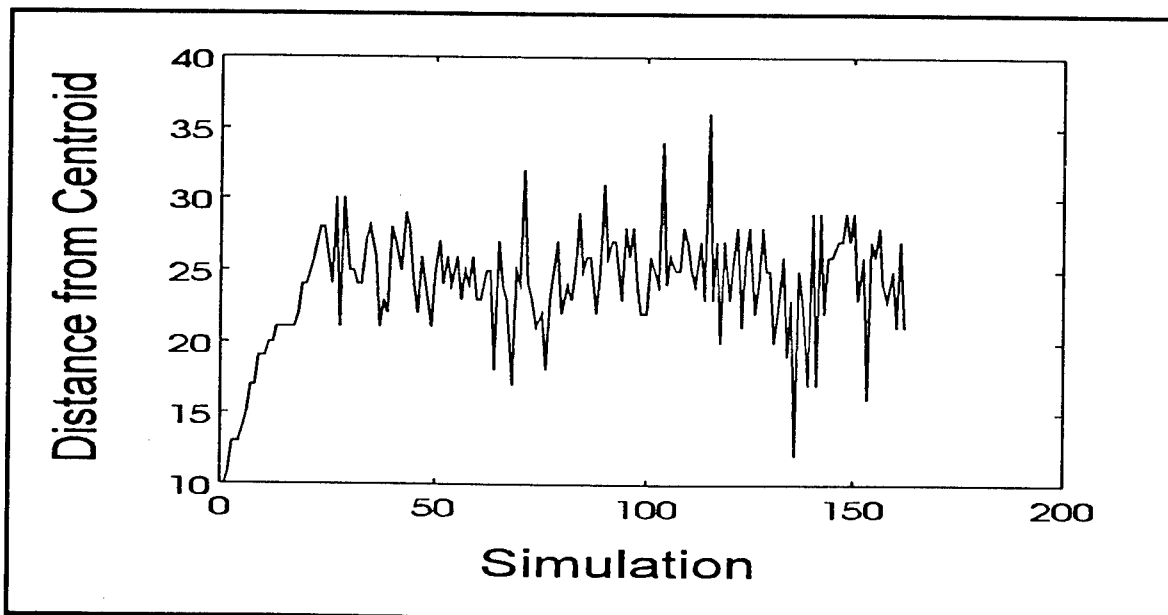


Figure 4.6.3. *Distance of All Simulations from the First Cluster Centroid.*

A final graphic displays the same type of information shown in Figure 4.6.3 for simulations and all categories.

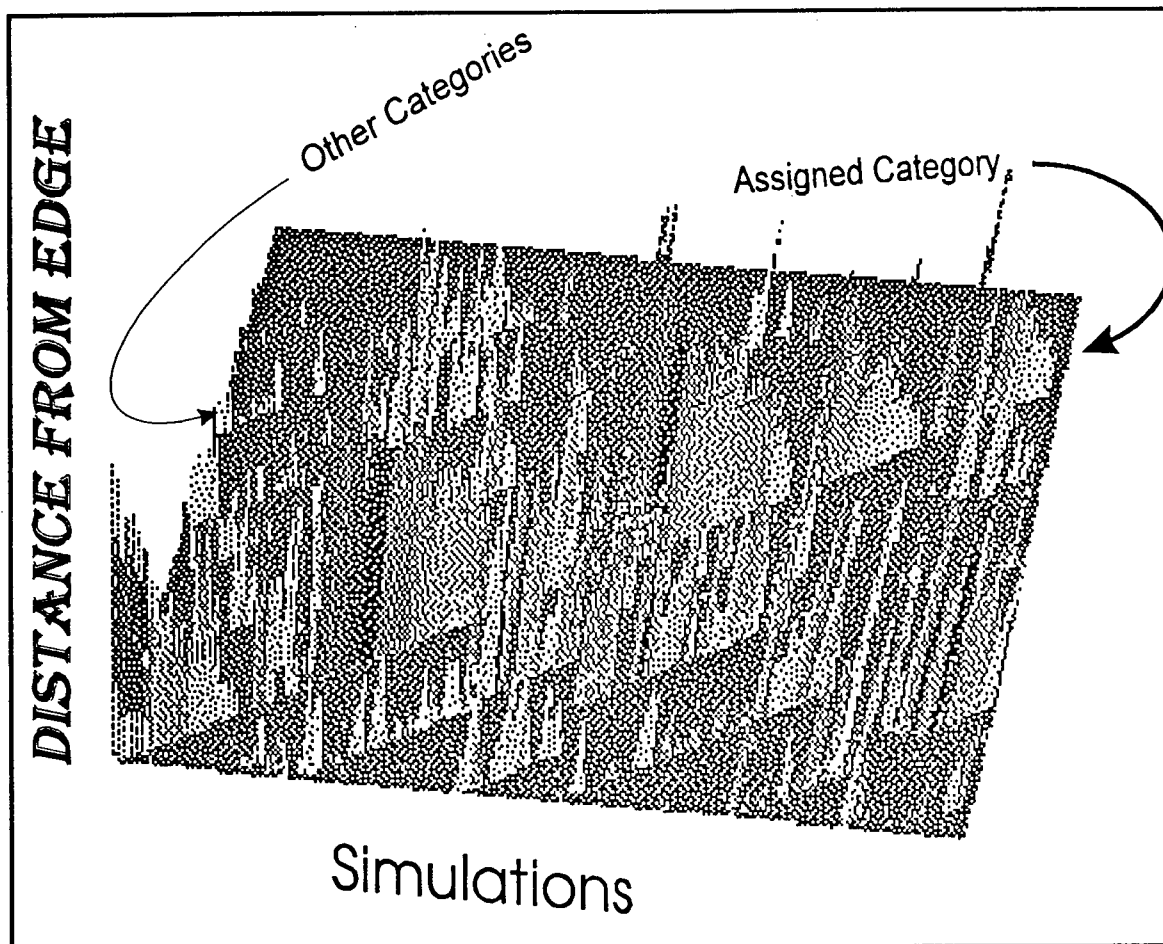


Figure 4.6.4. Distance of Each Simulation from the Center of its Category.

6.5. Results

From Figure 4.6.5 we see that, even without refinement, there is a structure to the selected simulations and that classes of metamodeling problems could be defined by these clusters.

7. REFERENCES

1. P. J. Antsaklis, "On Autonomy and Intelligence Control - from Defining Intelligent Control," *IEEE Control Systems Magazine*, April 1994.
2. A. Meystel, "On Intelligent Control, Learning and Hierarchies - from Defining Intelligent Control," *IEEE Control Systems Magazine*, April 1994.
3. D. C. Montgomery, *Design and Analysis of Experiments*, John Wiley & Sons, NY, 1991.
4. J. P. Norton, "An Introduction to Identification, Academic Press," New York, 1986.
5. D. A. Pam 5-11, *Verification, Validation,, and Accreditation of Army Models and Simulations*, HQ Department of the Army, 22 July 1993.
6. L.B. Anderson, et al, "SIMTAX, A Taxonomy for Warfare Simulation," Workshop report taken from the *Catalog of Wargaming and Military Simulation Models, 11th Edition*, Force Structure, Resource, and Assignment Directorate (J-8), The Joint Staff, Washington, DC 20318-8000, September 1989.
7. D. Caughlin, "A Metamodeling Approach to Model Abstraction," *Proc. 1994 IEEE Dual-Use Technologies & Application Conference*, May 1994.
8. D. Caughlin, "An Evaluation of Simulated Annealing for Modeling Air Combat Simulations," *Proc. 1994 IEEE Dual-Use Technologies & Application Conference*, May 1994.
9. *Catalog of Wargaming and Military Simulation Models, 11th Edition*, Force Structure, Resource, and Assignment Directorate (J-8), The Joint Staff, Washington, DC 20318-8000, September 1989.
10. V. Vemuri, *Modeling of Complex Systems*, Academic Press, 1978.
11. A. W. Naylor, and G. R. Sell, *Linear Operator Theory in Engineering and Science*, New York: Springer Verlag 1982.

CHAPTER 5

METAMODEL STRUCTURES

1. CHAPTER OUTLINE

1. Chapter Outline	5-1
2. Introduction.....	5-2
3. System Representation (Model Set)	5-3
3.1. Introduction	5-3
3.2. System Description	5-3
3.3. System Class	5-4
3.4. Model Structure	5-5
4. Predictor Models	5-6
4.1. Summary of Model Structures.....	5-6
4.1.1. Static Systems	5-6
4.1.2. Dynamic Systems.....	5-6
4.1.2.1. Linear Systems.....	5-6
4.1.2.2. Nonlinear Systems.....	5-8
4.2. Predictor Equations.....	5-9
4.2.1. Linear Time Invariant Predictors.....	5-10
4.2.1.1. Scalar Systems	5-10
4.2.1.2. Multivariable Systems	5-11
4.2.1.3. State Space Forms.....	5-12
4.2.2. Linear Time-Varying Predictors.....	5-14
4.2.2.1. Weighting Functions.....	5-14
4.2.2.2. Time Varying State-Space.....	5-14
4.2.3. Nonlinear Predictors	5-14
5. Probabilistic Models.....	5-15
5.1. Introduction	5-15
5.2. Linear Stochastic Models	5-15
5.2.1. Continuous Time Predictor	5-16
5.2.2. Discrete-Time Predictor.....	5-17
5.2.3. Steady State Solution.....	5-17
5.3. Nonlinear Stochastic Models	5-18

5.4.	Linear Ito Stochastic Models.....	5-18
5.4.1.	Brownian Motion	5-19
5.4.1.1.	Scalar Brownian Motion.....	5-19
5.4.1.2.	Vector Brownian Motion	5-20
5.4.2.	Stochastic Integrals.....	5-20
5.4.3.	Linear Stochastic Differential Equations.....	5-20
5.4.4.	Markov Processes.....	5-21
5.4.5.	Chapman-Kolmogorov Equation.....	5-22
5.5.	Nonlinear Ito Stochastic System Models.....	5-23
5.5.1.	Ito Stochastic Integrals and Differentials.....	5-23
5.5.1.1.	Properties of Ito Stochastic Integrals.....	5-23
5.5.2.	Nonlinear Ito Stochastic Differential Equations.....	5-24
5.5.3.	Properties of the Solution Process.....	5-25
5.5.4.	Ito Differential Rule.....	5-26
5.5.5.	Transition Probability Density.....	5-26
5.5.6.	Propagation of the Mean and Covariance.....	5-27
5.6.	Ito Stochastic Prediction	5-27
5.6.1.	Linear System Models	5-27
5.6.2.	Nonlinear System Models	5-27
6.	References.....	5-31

2. INTRODUCTION

Chapter 4 addressed the first eight steps of the metamodeling procedure that defined the prior knowledge. At this point, we have determined the purpose of the metamodel. In the definition of this purpose, we have identified the input and response that we are interested in and determined the important characteristics of these data. Also, for this purpose, we have defined the region of interest, selected validity measures, and specified the required validity.

This chapter begins the presentation of research to support Objective 2 and covers the models available to support decisions associated with **"Step 9: Postulate a metamodel."** The final aspect of the model selection, the order of the model, will be covered in the next chapter.

Methods for the complimentary steps ("Step 12: Fit the metamodel," and "Step 13: Access the validity of the model") are covered in Chapters 6 through 8. Selection of an experimental design (Step 10) is covered in Chapter 9, while the connection between these metamodeling methods and the experimental design (procedures to obtain data, generate, and validate the metamodel, Steps 11, 12, and 13) are presented in Chapter 10.

3. SYSTEM REPRESENTATION (MODEL SET)

3.1. Introduction

The first requirement of Objective 2 is to categorize the set of available metamodel structures. Recall the definition:

A model structure \mathcal{M} is defined as a differentiable mapping from a connected open subset \mathcal{D}_m of \mathbb{R}^d to a model set $\mathcal{M}(\theta)$, such that the gradients of the predictor functions are stable.

Note: \mathcal{D}_m has to be open (or be a differential manifold) so that the gradients of the prediction are well defined.

The completion of Step 9 (Postulate a metamodel) requires a number of selections concerning the system description (type and class), the model, determination of model order, and the criterion of fit.

One of the first decisions required is selection of the system description. In reality, all "real world" systems are complex, large scale interconnections of continuous-discrete, nonlinear, infinite-dimensional components.

One of the best techniques to represent and analyze these systems is a simulation. Mathematical simulation (as opposed to a more general definition of simulation where physical models or environments are used to represent the behavior) is a particular type of model structure that combines multiple representations to determine the system behavior. Following the framework in Chapter 3, a mathematical model is defined as the pair $\Sigma = (\mathbf{U}, \mathbf{B})$ with \mathbf{U} the universe of outcomes produced by the underlying phenomenon, and the \mathbf{B} , the behavior of the model. A dynamical system Σ is simply a triple $\Sigma = (\mathbf{T}, \mathbf{W}, \mathbf{B})$ with $\mathbf{T} \subseteq \mathbb{R}$ the time axis, \mathbf{W} the signal space, and $\mathbf{B} \subseteq \mathbf{W}^{\mathbf{T}}$ the behavior -- the set of all maps from \mathbf{T} to \mathbf{W} , a family of \mathbf{W} -valued time trajectories. A simulation of Σ is a procedure for selecting an arbitrary element of the behavior \mathbf{B} and defining an algorithm for computing it.

Metamodeling, as a method of abstraction, precludes this type of representation and requires a purely mathematical relationship. Mathematically, there are no closed form solutions to continuous-discrete, nonlinear, infinite-dimensional systems [1,2]. Consequently, these systems are usually approximated by finite dimensional linear or nonlinear systems.

3.2. System Description

Given that multiple model sets are available, the model structure that will define the behavior of the models must be determined. In the definition of the system description, the first selection concerns the system type. Here, the most basic questions must be addressed. How are the parameters described? Is the representation going to include dynamics or will it be static? Will the model contain latent variables? Are disturbances,

noise, and randomness accommodated? Table 5.3.1. defines possible descriptions and the types of models they represent.

Table 5.3.1. Parameter Descriptions and Model Representations.

PARAMETER DESCRIPTION	MODEL REPRESENTATION
Distributed parameter	Models described by partial differential equations Steady state Bifurcation points of nonlinear operators
Lumped parameter	Nonparametric Impulse response Frequency function Parametric Transfer function State space Valid linearizations

Given that the system can be described by lumped parameters, the next decision is between a parametric or nonparametric representation. Since nonparametric identification techniques typically require that a test signal be input to the system operating open loop, and simulations usually could not meet this requirement, these model structures will not be pursued further [3]. Table 5.3.2. defines possible selections for parametric system descriptions. Note that while static and dynamical models can both accommodate nonlinear and stochastic behavior, only the dynamical systems have time and trajectories associated with them. (Also, chaotic systems are deterministic.)

Table 5.3.2. System Descriptions.

TYPE	ALGEBRAIC STRUCTURE	RANDOMNESS	TIME	TRAJECTORY
Static	Linear	Deterministic		
Dynamic			Continuous	Chaotic
	Nonlinear	Stochastic	Discrete	Periodic
			Continuous-discrete (sampled data)	Aperiodic

3.3. System Class

In addition to the system description, the class of representation is also needed to define the overall system description. This class is defined by the model structure and

representation. Table 5.3.3 provides a list of the general system classes by their structure and representations [4,5].

Table 5.3.3. System Classes and Representations.

MODEL STRUCTURE	TYPICAL REPRESENTATION
Single-input-single-output (SISO)	ARX / ARMA / ... $A(t)y(t) = \frac{B(q)}{F(q)}u(t) + \frac{C(q)}{D(q)}e(t)$
Multiple-input-single-output (MISO)	
Multivariable systems Multiple-input-multiple-output (MIMO)	Input/State/Output: $\dot{x} = f(t, x(t), u(t), w(t); q)$ $y(t) = h(t, x(t), u(t), v(t); q)$

3.4. Model Structure

Once the system description and class have been determined, the next decision is selection of the model structure to use in describing the response of the system to the inputs (possibly including latent variables).

There are two general model structures, **predictor models** and **probabilistic models**. A predictor model only defines the predictor equation(s). Predictor models are models that specify the elements of the transfer function in terms of some parameter. The models generated from these structures are deterministic in nature. However, since the coefficients were generated via a minimization of some error criterion with assumed statistics, the coefficients will be random variables with an error distribution. Since the estimates are functions of these random variables, this distribution can be used to compute error bounds of the estimate.

A probabilistic model accommodates the fact that many systems are subject to known disturbances that are not (or cannot be) completely categorized. The statistics of the noises and disturbances can be included as random variables. Probabilistic models supplement the parametric description with a description of the density function (or moments) of the noise (disturbance) that acts on the system. The variables of the system being identified become functions of random variables. In these situations, different realizations of an experiment (simulation run) may not produce exactly the same results.¹ Consequently, the output of a probabilistic model is the conditional expected value and probability density functions (CPDF) of the variables. The following two subsections discuss these two model structures.

¹This is one reason that there are so many different predictor methods. An attempt has been made to accommodate the fact that a random variable with given moments has been approximated by a predictor model with a deterministic result.

4. PREDICTOR MODELS

4.1. Summary of Model Structures

Since all model structures are not appropriate for every system description, the available selections are dependent on the description of system we have selected.

4.1.1. Static Systems

Table 5.4.1 lists some of the available static model structures. Many of these structures would be most appropriate for pseudolinear regressions [4]. Also, any of the models described for a dynamical system could be used by neglecting the system dynamics and maintaining the input-state relationships.

Table 5.4.1. Static System Model Structures.

MODEL	EQUATION
Straight line	$y = a + bx$
Power	$y = bx^a$
Exponential	$y = ae^{Ax} + be^{Bx}$
Inverse type	$y = a + b/x$ $y = 1/(a + bx)$
Polynomial	$y = a + bx + cx^2 + dx^3$ $y = a_0 + a_1P_1(x_i) + a_2P_2(x_i) \cdots P_k(x_i)$ P a k^{th} order polynomial
Logistic	$y = k/(1 + be^{-ax})$

4.1.2. Dynamic Systems

We will consider three types of dynamic systems: linear time invariant, linear time-varying, and nonlinear. All nonlinear systems will be assumed to be Markov.

4.1.2.1. Linear Systems

There are a number of ways of defining the transfer functions associated with linear predictor models. First, the numerator and denominator of the transfer function can be given explicitly in either discrete or continuous time. This transfer function can also be converted into a frequency function that gives the frequency response of the transfer function. The transfer function can also be defined by the zeros and poles of the model. These descriptions are most appropriate for SISO systems.

MISO systems are best represented by a state space or polynomial format that explicitly defines the coefficients of each of the input and output terms. MIMO systems are most amenable to the state space format. This format also has the most flexibility in defining the relationship to latent variables. Latent variables (that are not past values of the input

or output) can also be defined in the polynomial format by augmenting the input-output relationships.

4.1.2.1.1. Linear Time Invariant Systems

Since we are most concerned with MIMO and MISO systems, the polynomial and state space formats will be emphasized. In addition to the **AR** and **MA** representations given in Chapter 3, the following descriptions (Table 5.4.2) add the capability to identify the input functions in an input-output relationship where the **X** refers to the input (exogenous) variable [4]. (Unfortunately, the definitions below are not universal. For example, some authors refer to the ARX model structure as an ARMA model with the moving average defined over the input rather than the error.)

Table 5.4.2. Linear Time-Invariant (or Stationary) System Model Structures.

TYPE	REPRESENTATION
Finite Impulse Response (FIR)	$y(t) = b_1u(t-1) + \dots + b_nbu(t-n_b)$
Autoregressive -- (AR)	$y(t) + a_1y(t-1) + \dots + a_nay(t-n_a) = + e(t)$
Equation Error Model Autoregressive -- (ARX)	$y(t) + a_1y(t-1) + \dots + a_nay(t-n_a) = b_1u(t-1) + \dots + b_nbu(t-n_b) + e(t)$
Autoregressive Moving Average (ARMA)	$y(t) + a_1y(t-1) + \dots + a_nay(t-n_a) = + e(t) + c_1e(t-1) + \dots + c_nce(t-n_c)$
Moving Average (MA)	$y(t) = + e(t) + c_1e(t-1) + \dots + c_nce(t-n_c)$
Autoregressive Moving Average (ARMAX)	$y(t) + a_1y(t-1) + \dots + a_nay(t-n_a) = b_1u(t-1) + \dots + b_nbu(t-n_b) + e(t) + c_1e(t-1) + \dots + c_nce(t-n_c)$
Generalized Least Squares (ARARX)	$A(q)y(t) = B(q)u(t) + \frac{1}{D(q)}e(t)$
Extended Matrix Model (ARARMAX)	$A(q)y(t) = B(q)u(t) + \frac{C(q)}{D(q)}e(t)$
Output Error Model	$y(t) = \frac{B(q)}{F(q)}u(t) + e(t)$
Box-Jenkins Model	$y(t) = \frac{B(q)}{F(q)}u(t) + \frac{C(q)}{D(q)}e(t)$
State Space Model	Directly parametrized innovations form of $\dot{\mathbf{x}} = \mathbf{F}(\theta) \mathbf{x}(t) + \mathbf{G}(\theta) \mathbf{u}(t) + \mathbf{L}(\theta) \mathbf{w}(t)$

In the above table, q^{-1} is the backward shift operator defined as $q^{-1}u(t) = u(t-1)$. The polynomials are defined as:

$$A(q) = 1 + a_1 q^{-1} + \dots + a_{n_a} q^{-n_a}$$

$$B(q) = b_1 q^{-1} + \dots + b_{n_b} q^{-n_b}$$

$$C(q) = 1 + c_1 q^{-1} + \dots + c_{n_c} q^{-n_c}$$

$$D(q) = 1 + d_1 q^{-1} + \dots + d_{n_d} q^{-n_d}$$

$$F(q) = 1 + f_1 q^{-1} + \dots + f_{n_f} q^{-n_f}$$

Since we are representing the same system with the different structures, there is a direct correspondence between them. This correspondence is discussed in the next chapter on model order using the ARX model as an example.

4.1.2.1.2. Linear Time Varying Systems

Low frequency disturbances, such as offsets, trends, drift, and periodic variations can be accommodated either by removing the disturbance from the data or by modifying the noise model to accommodate it. This modification includes an integrator and leads to the autoregressive integrated moving average model shown in Table 5.4.3. Other forms most suitable for time-varying systems are the weighting function and time varying state space.

Table 5.4.3. Linear Nonstationary and Time-Varying and Nonlinear System Model Structures.

STRUCTURE	TYPE	REPRESENTATION
Linear nonstationary time series models	Autoregressive integrated moving average model (ARIMAX)	
Linear time-varying	Weighting function	$y(t) = \sum_{s=-\infty}^{t-1} g(t,s)u(s) + v(t)$
	Time-varying state space	$\dot{x} = F(t,\theta)x(t,\theta) + G(t,\theta)u(t) + L(t,\theta)w(t)$

4.1.2.2. Nonlinear Systems

The following table shows the two basic approaches for defining nonlinear system structures. We can either work with the nonlinear system directly or we can attempt to approximate the nonlinear system with a stationary or time-varying linear system. The motivation for linearization lies in the fact that closed form or straight forward numerical identification methods exist for these systems while nonlinear methods primarily rely on minimization of some cost function (see Chapter 7). The process of linearization is based on perturbation methods and a small-signal approximation.

Table 5.4.4. Nonlinear System Model Structures.

STRUCTURE	TYPE	REPRESENTATION
Nonlinear	Weighting function	$y(t) = \sum_{s=-\infty}^{t-1} g(t,s)u(s) + v(t)$
	State space	$\dot{x} = f(t, x(t), u(t), w(t); q)$ $y(t) = h(t, x(t), u(t), v(t); q)$
	Chaotic	
Linearization		Stationary or time-varying

Assume that the actual system behavior is described by the state space equations above. The method of small signals is reflected in the assumption that $x(t)$ and $u(t)$ will always be close to some reference value x_0 and u_0 . And, furthermore, that these values are an equilibrium point: $\dot{x} = f(t, x(t), u(t), w(t); q) = 0$. Since $x(t)$ and $u(t)$ are always close to x_0 and u_0 , they can be written as $x(t) = x_0 + \delta x(t)$ and $u(t) = u_0 + \delta u(t)$ with $\delta x(t)$ and $\delta u(t)$ assumed small. Therefore, we can perform a Taylor series expansion of the actual system about the point x_0 and u_0 leading to:

$$\frac{d}{dt}(x_0 + \delta x(t)) = f(x_0 + \delta x(t), u_0 + \delta u(t))$$

with, for each of the states i , gives:

$$\delta \dot{x}_i = f_x(x_0, u_0) \delta x(t) + f_u(x_0, u_0) \delta u(t)$$

where the Jacobian f_x is defined as (with f_u defined similarly):

$$f_x = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \dots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \dots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \dots & \frac{\partial f_n}{\partial x_n} \end{pmatrix}$$

Since $f(x_0, u_0)$ is an equilibrium point, the result has the same form with the system and input functions defined by the Jacobians given above.

4.2. Predictor Equations

Static systems can be either linear or nonlinear. The predictor equations for static models are the actual input-output map that comes from the selected representation and are similar to those representing dynamical systems. Also, static models can be set up using dynamical model structures with a zero state transition.

For dynamic systems, system identification requires the ability to use the model structure to predict the output of the model. The differences between this prediction and the actual data are then used to arrive at the parameter set which minimizes the error.

As the complexity of the system description increases, the flexibility of the predictor form decreases. Linear time-invariant systems have multiple forms that can be used ranging from a pure transfer function to state space. Linear time-varying systems are restricted to weighting function and state space forms. Nonlinear systems (that are not approximated by linearization or perturbation) are basically restricted to state space descriptions that explicitly calculate responses.

4.2.1. Linear Time Invariant Predictors

Assume that the system description is given by the following form with an invertible noise model $H(q)$:

$$y(t) = G(q)u(t) + H(q)e(t)$$

4.2.1.1. Scalar Systems

For a scalar system, a one-step-ahead prediction (conditional expectation) of $\hat{y}(t)$ can be determined from the past values of the noise, and past and current values of inputs and outputs.

$$\hat{y}(t|t-1) = H^{-1}(q)G(q)u(t) + (1 - H^{-1}(q))y(t)$$

Consequently, a predictor for the general model structure:

$$A(q)y(t) = \frac{B(q)}{F(q)}u(t) + \frac{C(q)}{D(q)}e(t)$$

is:

$$\hat{y}(t|\theta) = \frac{D(q)B(q)}{C(q)F(q)}u(t) + \left[1 - \frac{D(q)A(q)}{C(q)}\right]y(t)$$

We can also write the above equation as a recursion:

$$C(q)F(q)\hat{y}(t|\theta) = D(q)B(q)u(t) + F(q)[C(q) - D(q)A(q)]y(t)$$

and using this recursion, the prediction error $\varepsilon(t, \theta) = y(t) - \hat{y}(t|\theta)$ can be written:

$$\varepsilon(t, \theta) = \frac{D(q)}{C(q)} \left[A(q)y(t) - \frac{B(q)}{F(q)}u(t) \right]$$

defining the intermediate variables

$$v(t, \theta) = \frac{B(q)}{F(q)} u(t)$$

$$g(t, \theta) = A(q)y(t) - v(t, \theta)$$

the prediction error can be written:

$$\varepsilon(t, \theta) = \frac{D(q)}{C(q)} g(t, \theta)$$

With the following "state vector"

$$\varphi(t, \theta) = \begin{bmatrix} -y(t-1), \dots, -y(t-n_a), \\ u(t-1), \dots, u(t-n_b), \\ -v(t-1, \theta), \dots, -v(t-n_f, \theta), \\ -\varepsilon(t-1, \theta), \dots, -\varepsilon(t-n_c, \theta), \\ -g(t-1, \theta), \dots, -g(t-n_d, \theta) \end{bmatrix}^T$$

and parameter vector

$$\theta = [a_1, \dots, a_{n_a}, b_1, \dots, b_{n_b}, f_1, \dots, f_{n_f}, c_1, \dots, c_{n_c}, d_1, \dots, d_{n_d}]^T$$

This predictor can also be rewritten in a pseudolinear form $\hat{y}(t|\theta) = \theta^T \varphi(t)$.

With θ confined to $\mathcal{D} = \{\theta \mid F(z) \cdot C(z) \text{ has no zeros on or outside the unit circle}\}$ (stable) the parameterization of the predictor above meets the definition of a model structure given in the introduction.

4.2.1.2. Multivariable Systems

Consider the situation where the input is an m -dimensional vector, and the output is a p -dimensional vector. In this case, the term $\frac{B(q)}{F(q)}$ has no meaning and a **matrix fraction description** (MFD) is required.

For these structures, we can still define $A(q) = 1 + a_1 q^{-1} + \dots + a_{n_a} q^{-n_a}$. However, $A(q)$ is now a $p \times p$ matrix; other matrices follow appropriately.

While the system is still given by $y(t) = G(q)u(t) + H(q)e(t)$ with a general model structure of $y(t) = A^{-1}(t)F^{-1}(q)B(q)u(t) + A^{-1}(t)D^{-1}(q)C(q)e(t)$, the situation is not as simple as it may seem. This description is not unique, and there are many, many issues associated with the form, order, and identifiability of the representation. These issues will

be addressed in subsequent chapters. An excellent discussion of the properties of the MFDs (155 pages) is available in Chapter 6 of [6].

4.2.1.3. State Space Forms

The state space form is given by: $\dot{x} = F(t)x(t) + G(t)u(t) + L(t)w(t)$ with continuous measurements: $y(t) = H(t)x(t) + v(t)$ and the following disturbance assumptions:

$$E\{w(t)w^T(t+\tau)\} = Q(t)\delta(\tau)$$

$$E\{v(t)v^T(t+\tau)\} = R(t)\delta(\tau)$$

$$E\{w(t)v^T(t+\tau)\} = S(t)\delta(\tau)$$

While both continuous and discrete prediction equations will be provided for probabilistic models, we will only discuss the discrete version of the predictor model. Time invariant descriptions are obtained by suppressing the time dependency. Using the following relationships we can derive a discrete time predictor model that will accommodate the fact that there is measurement and process noise -- the directly parameterized innovations form. Let:

$$\begin{aligned} A(t, \theta) &= \Phi(t, t_0; \theta) = e^{F(t-t_0)} \\ B(t, \theta) &= \int_{t_i}^{t_{i+1}} \Phi(t_{i+1}, \tau; \theta) G(\tau) d\tau \\ M(t, \theta) &= \int_{t_i}^{t_{i+1}} \Phi(t_{i+1}, \tau; \theta) L(\tau) d\tau \end{aligned}$$

then we can convert the continuous system to a fully discrete system:

$$x_{t_{i+1}} = A(t_i, \theta) x(t_i) + B(t_i, \theta) u(t_i) + M(t_i, \theta) w_d(t_i)$$

If $C(t_i, \theta) = H(t_i) \Phi(t_i, t_i)$, then the measurement equation supporting this system, allowing for noncoincident sampling and control is:

$$y_{t_i} = C(t_i, \theta) x(t_i) + \left[H(t_i) \int_{t_i}^{t_i} \Phi(t_i, \tau) G(\tau) d\tau \right] u(t_i) + \left[H(t_i) \int_{t_i}^{t_i} \Phi(t_i, \tau) Q(\tau) d\tau \right] + v_d(t_i)$$

Also, the disturbances must be converted to discrete time equivalents. The process noise becomes²:

$$E\{w_d(t_i)w_d^T(t_i)\} = Q_d(t_i) = \int_{t_i}^{t_{i+1}} \Phi(t_{i+1}, \tau) L(\tau) Q(\tau) L^T(\tau) \Phi(t_{i+1}, \tau) d\tau$$

In converting the measurement noise to the discrete case, the correlation time of the sensor must be considered [7]. Assuming that time correlation is exponential with a correlation time τ_c , and that $\tau_c \ll$ System time constants³,

$$E\{v_d(t_i)v_d^T(t_i)\} = R_d(t_i) = \frac{2}{T} [\tau_c E\{vv^T\}]$$

and

$$E\{w_d(t_i)v_d^T(t_i)\} = S_d(t_i) = \frac{2}{T} [\tau_c E\{wv^T\}]$$

If we consider a steady state solution for the time invariant case, the discrete time predictor (conditional expected value) for this system is the classical steady state Kalman filter:

$$\hat{x}(t_{i+1}, \theta) = A(\theta) \hat{x}(t_i, \theta) + B(\theta) u(t_i) + K(\theta) [y(t_i) - C(\theta) \hat{x}(t_i, \theta)]$$

and

$$y(t_i, \theta) = C(\theta) \hat{x}(t_i, \theta)$$

with

$$K(\theta) = [A(\theta) \bar{P}(\theta) C^T(\theta) + S_d(\theta)] [C(\theta) \bar{P}(\theta) C^T(\theta) + R_d(\theta)]^{-1}$$

and

$$P(\theta) = A(\theta) \bar{P}(\theta) A^T(\theta) + Q_d(\theta) - [A(\theta) \bar{P}(\theta) C^T(\theta) + S_d(\theta)] \cdot [C(\theta) \bar{P}(\theta) C^T(\theta) + R_d(\theta)]^{-1} \cdot [A(\theta) \bar{P}(\theta) C^T(\theta) + S_d(\theta)]$$

where $\bar{P}(\theta)$ is the positive semidefinite solution (to the discrete algebraic Ricatti equation). As given above, knowledge of the disturbance properties is required which makes this a probabilistic model. However, if we do not try to parameterize the process and noise descriptions, but do parameterize and identify the Kalman Gain $K(\theta)$, the result is a predictor model called the **directly parametrized innovations form**:

$$x(t_{i+1}, \theta) = A(\theta) \hat{x}(t_i, \theta) + B(\theta) u(t_i) + K(\theta) [e(t_i)]$$

²This integral is approximately $Q_d(t_i) \approx TL(t_i)Q(t_i)L^T(t_i) \approx M(t_i)\frac{Q(t_i)}{T}M^T(t_i) \approx \frac{Q(t_i)}{T}$

³ $R_d \approx R/T$

Comparing this to the general model structure $y(t) = G(q)u(t) + H(q)e(t)$, we see that

$$G(q, \theta) = C(\theta)[qI - A(\theta)]^{-1} B(\theta)$$

$$H(q, \theta) = C(\theta)[qI - A(\theta)]^{-1} K(\theta) + I$$

It should be noted again that the multivariable descriptions are not unique. There are, however, defined structures within each description that can be directly related to each other. These are called canonical forms [6, 7, 8]. The next chapter on the determination of model order discusses two of the canonical forms: the controller canonical form and the observer canonical form.

Consider the parameterization of the state space predictor above. Assume that $A(\theta)$, $B(\theta)$, $K(\theta)$, and $C(\theta)$ are differentiable with respect to θ . Suppose that $\theta \in \mathcal{D}$ with $\mathcal{D} = \{\theta | \text{all eigenvalues of } A(\theta) - K(\theta)C(\theta) \text{ are inside the unit circle}\}$ (stable); then this parameterization meets the definition of a model structure.

4.2.2. Linear Time-Varying Predictors

There are two general models listed for handling time varying systems: the weighting function and the time-varying state space.

4.2.2.1. Weighting Functions

Models for use with a weighting function are the same models that are used for time-invariant systems except that the weighting function, $G(t, q)$, is time-varying.

4.2.2.2. Time Varying State-Space

Time varying state space models are similar to the time invariant state models with the exception of the time index on the coefficients.

$$\dot{x} = F(t, \theta)x(t, \theta) + B(t, \theta)u(t) + G(t, \theta)w(t)$$

Closed form solutions similar to the discrete steady state Kalman filter do not exist for the time varying case.

4.2.3. Nonlinear Predictors

With respect to general dynamical nonlinear models, the situation is far too flexible. The output may be a function of all of the past inputs and outputs, yet we are going to represent this system with a finite number of parameters. Usually considerable insight is required to effectively use a nonlinear model type.

Nonlinear models can be written in a pseudolinear form:

$$\hat{y}(t|\theta) = \theta^T \phi(t)$$

where θ is the vector of unknown coefficients and ϕ contains the nonlinear combinations (functions) of the input data. Although the structure looks static, dynamics can be included in the nonlinear model by including nonlinear combinations of past data.

Systems with linear dynamics and static input nonlinearities can be handled by redefining input of the system to exclude this nonlinearity (Hammerstein model). With this new definition, the system can be identified by a linear model [3].

If we want to explicitly consider system dynamics, there are two options: a nonlinear state space models and a simulation model.

$$\begin{aligned}\dot{x} &= f(t, x(t), u(t); q) \\ y(t) &= h(t, x(t), u(t); q)\end{aligned}$$

A simulation model, not to be confused with a simulation as a system description, simulates $\hat{y}(t|\theta)$ by simulating a noise free state space model using actual inputs.

5. PROBABILISTIC MODELS

5.1. Introduction

Models for probabilistic descriptions will be limited to the state space form. While transfer function and matrix fraction descriptions are limited to linear time-invariant systems, a state space system does not share this restriction. This form also allows the combination of a continuous system with discrete measurements (a sampled-data system) to more closely match real systems.

The subsection covers three types of probabilistic models. The first type of model is a linear stochastic model developed by assuming a white noise approximation. The second model is a linear Ito stochastic model based on the correct description of the noise as Brownian motion with an Ito stochastic description, and the final model is a full nonlinear Ito stochastic model.

Since the description is limited to state space forms, the following discussion concentrates on the predictor equations for the parameter estimate.

5.2. Linear Stochastic Models

Linear stochastic system modeling results in the following time varying model driven by known inputs and white noise [2]:

$$\dot{x} = F(t)x(t) + G(t)u(t) + L(t)w(t)$$

starting from a Gaussian $x(t_0)$ with a known mean \hat{x}_0 and covariance P_0 . Average performance can often be described by this simple stochastic differential equation sometimes referred to as Langevin's equation [9].

The solution to this equation is:

$$x(t) = \Phi(t, t_0)x(t_0) + \int_{t_0}^t \Phi(t, \tau)B(\tau)u(\tau)d\tau + \int_{t_0}^t \Phi(t, \tau)G(\tau)d\beta(\tau)$$

where $\Phi(t, t_0)$ is the transition matrix associated with $F(t)$.

In discrete time, the model and solution become:

$$x_{t_{i+1}} = A(t_i) x(t_i) + B(t_i) u(t_i) + M(t_i)w_d(t_i)$$

The continuous model is supported by a linear measurement corrupted by additive white noise:

$$y(t) = H(\theta) x(t) + v(t)$$

or with noncoincident sampling and control in discrete time:

$$y_{t_i} = C(\theta) x(t_i) + \left[H(t_i) \int_{t_i}^{t_{i+1}} \Phi(t_{i+1}, \tau)G(\tau)d\tau \right] u(t_i) + \left[H(t_i) \int_{t_i}^{t_{i+1}} \Phi(t_{i+1}, \tau)Q(\tau)d\tau \right] + v_d(t_i)$$

(See "State Space Forms" above for the discrete time conversion factors and noise covariances.)

Since the solution of these systems is a stochastic process with many potential realizations, it is best to characterize the system by the expected value of its moments (mean, variance, etc.) The optimal (minimum mean square error, unbiased, consistent) predictor for this system is the classical Kalman-Bucy Filter.

The output of the model is $y(t)$. Define $z(t)$ or $z(t_i)$ as the measurements available for the prediction. The measurement model remains the same as the output model.

5.2.1. Continuous Time Predictor

The continuous-time predictor consists of the following set of equations:

State estimate:

$$\dot{\hat{x}}(t) = F(t)\hat{x}(t) + G(t)u(t) + K(t)[z(t) - H(t)\hat{x}(t)]$$

Filter gain calculation:

$$K(t) = [P(t)H^T(t) + L(t)S(t)]R^{-1}(t)$$

Error covariance propagation (Riccati equation):

$$\dot{P}(t) = F(t)P(t) + P(t)F^T(t) + L(t)Q(t)L^T(t) - K(t)R(t)K^T(t)$$

5.2.2. Discrete-Time Predictor

The discrete-time predictor includes an additional step beyond those required for the continuous filter. Given a state and covariance estimate, those estimates are first extrapolated to the next time step (without taking a measurement). At the next time step, a measurement is taken and the estimates are updated yielding (t_i^+) .

State estimate extrapolation:

$$\begin{aligned}\hat{x}(t_i) &= A(t_{i-1})\hat{x}(t_{i-1}) + \int_{t_{i-1}}^{t_i} \Phi(t_i, \tau)G(\tau)u(\tau)d\tau \\ &= A(t_{i-1})\hat{x}(t_{i-1}) + B(t_{i-1})u(t_{i-1})\end{aligned}$$

Error covariance extrapolation:

$$P(t_i) = A(t_{i-1})P(t_{i-1})A^T(t_{i-1}) + Q_d(t_{i-1})$$

Filter gain calculation:

$$K(t_i) = P(t_i)C^T(t_i)[O(t_i)]^{-1}$$

with:

$$O(t_i) = C(t_i)P(t_i)C^T(t_i) + R_d(t_i)$$

State estimate update:

$$\hat{x}(t_i^+) = \hat{x}(t_i) + K(t_i)[z(t_i) - C(t_i)\hat{x}(t_i)]$$

Error covariance update:

$$P(t_i^+) = [I - K(t_i)C(t_i)]P(t_i)$$

5.2.3. Steady State Solution

If the system and measurement dynamics are linear, constant coefficient equations, the disturbance and noise are stationary (Q, R, S not function of time), the filtering process will reach a steady state where the value of P is constant. For these conditions the continuous Riccati equation becomes an algebraic relationship:

$$\dot{P} = FP + PF^T + LQL^T - KRK^T = 0$$

In this case, the rate at which uncertainty increases is just balanced by the new information available. The positive semidefinite solution of the continuous algebraic Riccati equation is used to calculate the constant filter gain. (The discrete steady state solution and algebraic Riccati equation was provided in the presentation of the directly parameterized innovations form.)

State augmentation can be used to handle situations where the process disturbances or measurement noise is correlated (see [1] or [10] for further discussion).

5.3. Nonlinear Stochastic Models

If we want to explicitly consider system dynamics, there is one option: a nonlinear space simulation model.

$$\begin{aligned}\dot{x} &= f(t, x(t), u(t), w(t); q) \\ y(t) &= h(t, x(t), u(t), v(t); q)\end{aligned}$$

In this case, since we want a probabilistic predictor, we include the process and measurement noise descriptions. Given the statistics of the disturbances, we compute an ensemble of trajectories using Monte Carlo techniques. From this ensemble, we compute expected values and the their distributions

A simulation model, not to be confused with a simulation as a system description, disregards the process noise, $w(t)$, and simulates $\hat{y}(t|\theta)$ by simulating a noise free state space model using actual inputs.

5.4. Linear Ito Stochastic Models

As reasonable as this model seemed, it was not completely suitable. Although other models may be derived from these Langevin type equations, the Markovian description is typically lost. With this loss, complete knowledge of the probability density functions is required to determine system properties. This information is usually not available.

In the development of the model, $w(t)$ has been considered as the derivative of a process with independent, stationary increments. Actually, the term $w(\cdot, \cdot)$ is the hypothetical derivative of Brownian motion (or the Wiener process). A hypothetical derivative is used because the correct solution could not be properly developed with ordinary Riemann integrals.

Linear stochastic differential equations can be properly developed through the use of Wiener stochastic integrals [1]. A Wiener stochastic integral can be defined for a nonrandom $A(\cdot)$ by means of a mean square limit:

$$\begin{aligned}I(t, \cdot) &= \int_{t_0}^t A(\tau) d\beta(\tau, \cdot) \\ &= \text{l.i.m.}_{N \rightarrow \infty} \int_{t_0}^t A_N(\tau) d\beta(\tau, \cdot) \\ &= \text{l.i.m.}_{N \rightarrow \infty} \sum_{i=0}^{N-1} A_N(t_i) [\beta(\tau_{i+1}, \cdot) - \beta(\tau_i, \cdot)]\end{aligned}$$

where N is the number of time cuts made in $[t_0, t]$ and $A_N(\tau) = A(t_i)$ for all $\tau \in [t_i, t_{i+1})$.

5.4.1. Brownian Motion

5.4.1.1. Scalar Brownian Motion

As a stochastic process, the stochastic integral is a Brownian motion process. Scalar Brownian motion, for example, is defined to be a process with independent increments that are Gaussian such that for any $x(t_1)$ and $x(t_2)$ in the time set T of interest:

$$E\{\beta(t_2) - \beta(t_1)\} = 0$$
$$E\{[\beta(t_2) - \beta(t_1)]^2\} = \int_{t_1}^{t_2} q(\tau) d\tau$$

with probability 1 (and $\beta(t_0) = 0$).

- Markov - true of any process with independent increments
- Continuous everywhere with probability one (or, "almost surely," i.e., all sample functions are continuous except possibly a set of total probability zero) and also in the mean square sense (or, in the "quadratic mean")
- Nondifferentiable everywhere with probability one and in the mean square sense
- Not of bounded variation with probability one and in the mean square sense
- $E\{\beta(t_i)\beta(t_j)\} = E\{\beta(t_i)^2\}$ for $t_j \geq t_i$.

Since $\beta(\cdot, \cdot)$ is a zero mean process with independent increments, it can be shown to be martingale.

A martingale is a stochastic process $x(\cdot, \cdot)$ for which $E\{|x(t)|\}$ is finite for all admissible (t) and $E\{x(t_i)|x(t_{i-1}), \dots, x(t_0)\} = x(t_{i-1})$ for any sequential times $(t_0), (t_1), \dots, (t_i)$. Or in continuous time, if $x(\cdot, \cdot)$ is defined over some interval T , then $E\{x(t)|x(\tau), t_0 \leq \tau \leq t' \leq t\} = x(t')$. This can be written more rigorously as $E\{x(t)|\mathcal{F}_{t'}\} = x(t')$, where \mathcal{F} is the minimal σ -algebra generated by $\{x(\tau), t_0 \leq \tau \leq t' \leq t\}$.

It can be proven that if $x(\cdot, \cdot)$ is a martingale that it is continuous with probability one and covariance $E\{[x(t_2) - x(t_1)]^2 | \mathcal{F}_{t_1}\} = (t_2 - t_1)$, then $x(\cdot, \cdot)$ is a Brownian motion with unit diffusion.

5.4.1.2. Vector Brownian Motion

Vector Brownian motion is a zero mean vector process $\beta(\cdot, \cdot)$ that has independent Gaussian increments with:

$$E\{[\beta(t_2) - \beta(t_1)][\beta(t_2) - \beta(t_1)]^T\} = \int_{t_1}^{t_2} Q(t)dt$$

that is continuous but nondifferentiable with probability one (almost surely) in the mean square sense.

Also, Brownian motion has the Levy oscillation property (quadratic variation property): If $\beta(\cdot, \cdot)$ is a unit-diffusion Brownian motion and $\{t_0, t_1, \dots, t_N = t_f\}$ is a partition of the interval $[t_0, t_f]$, then

$$\lim_{\max|t_{i+1} - t_i| \rightarrow 0} \sum_{i=0}^{N-1} [\beta'(t_{i+1}) - \beta'(t_i)]^2 = (t_f - t_0)$$

where the limit exists both in the mean square sense and with probability one. Therefore, $[d\beta'(t)]^2 = dt$. This will allow the evaluation of $I(t, \cdot)$ independent of the choice point in the interval (as opposed to the Stratonovich definition of the stochastic integral [11]).

5.4.2. Stochastic Integrals

The stochastic integral is a Brownian motion process with rescaled diffusion:

$$E\{[I(t_2) - I(t_1)][I(t_2) - I(t_1)]^T\} = \int_{t_1}^{t_2} A(\tau)Q(\tau)A^T(\tau)d\tau$$

Stochastic differentials ($dI(t, \cdot) = A(t)d\beta(t, \cdot)$) are properly defined as functions, that when integrated over appropriate limits, yield stochastic integrals.

5.4.3. Linear Stochastic Differential Equations

Therefore, the properly defined linear stochastic differential equation is:

$$dx(t) = F(t)x(t)dt + B(t)u(t)dt + G(t)d\beta(t)$$

where $\beta(\cdot, \cdot)$ is of diffusion strength $Q(t)$ for all t of interest given by $E\{d\beta(t)d\beta^T(t)\} = Q(t)dt$.

The solution to this stochastic differential equation is the stochastic process $x(\cdot, \cdot)$ given by:

$$x(t) = \Phi(t, t_0)x(t_0) + \int_{t_0}^t \Phi(t, \tau)B(\tau)u(\tau)d\tau + \int_{t_0}^t \Phi(t, \tau)G(\tau)d\beta(\tau)$$

with $\Phi(t, t_0)$ the state transition matrix associated with F .

In general, characterization of this process requires the joint probability density (or distribution if the density cannot be assumed to exist) of $x(t_1), x(t_2), \dots, x(t_N)$ for any number N of time cuts in the interval of interest by repeated application of Bayes rule. If $x(\cdot, \cdot)$ is a Markov process, however, specification of the transition probability densities completely specifies the joint densities.

5.4.4. Markov Processes

Let $x(\cdot, \cdot)$ be a vector stochastic process with conditional probability distribution

$$F_{x(t_i)|x(t_{i-1}),x(t_{i-2}),\dots,x(t_j)}(\xi_i|\xi_{i-1},\xi_{i-2},\dots,\xi_j)$$

given that $x(t_{i-1}, \omega_k) = \xi_{i-1}$, etc. If, for any countable choice of values of i and j ,

$$F_{x(t_i)|x(t_{i-1}),x(t_{i-2}),\dots,x(t_j)}(\xi_i|\xi_{i-1},\xi_{i-2},\dots,\xi_j) = F_{x(t_i)|x(t_{i-1})}(\xi_i|\xi_{i-1})$$

then $x(\cdot, \cdot)$ is a Markov process. This property for stochastic processes is analogous to the ability to define a state for deterministic systems.

Suppose that there are N possible discrete state values, indexed by the integer j , that a system of interest can assume at any given time. Associated with each state value j , we can define a state probability $p_j(t_i)$ as the probability that the system will be in state j at time t_i . These separate state probabilities can be arrayed as a vector:

$$p(t_i) = \begin{bmatrix} p_1(t_i) \\ p_2(t_i) \\ \vdots \\ p_N(t_i) \end{bmatrix} = \begin{bmatrix} P(\{\omega: x(t_i, \omega) = 1\}) \\ P(\{\omega: x(t_i, \omega) = 2\}) \\ \vdots \\ P(\{\omega: x(t_i, \omega) = N\}) \end{bmatrix}$$

If the system has the Markov property, then the probability of a transition from state k to state j by the next discrete time of interest is a function of j and k only, and not of the history of the system. Therefore, a state transition probability for a discrete-state Markov process can be defined as $T_{jk}(t_{i+1}, t_i) = P\{x(t_{i+1}) = j | x(t_i) = k\}$.

Given that the N state values are mutually exclusive and collectively exhaustive $\sum_{j=1}^N T_{jk}(t_{i+1}, t_i) = 1$ for $k = 1, 2, \dots, N$. This leads to a state transition probability matrix:

$$T(t_{i+1}, t_i) = \begin{bmatrix} T_{11}(t_{i+1}, t_i) & T_{12}(t_{i+1}, t_i) & \cdots & T_{1N}(t_{i+1}, t_i) \\ T_{21}(t_{i+1}, t_i) & T_{22}(t_{i+1}, t_i) & \cdots & T_{2N}(t_{i+1}, t_i) \\ \vdots & \vdots & \ddots & \vdots \\ T_{N1}(t_{i+1}, t_i) & T_{N2}(t_{i+1}, t_i) & \cdots & T_{NN}(t_{i+1}, t_i) \end{bmatrix}$$

where each column k depicts the probability of transitioning into state 1, state 2, etc., from any given state k . Each row j relates the probabilities of reaching state j from state 1, state 2, etc. With the transition matrix, the state probabilities at each time are expressed as $p(t_{i+1}) = T(t_{i+1}, t_i)p(t_i)$ with the j^{th} component $p_j(t_{i+1}) = \sum_{k=1}^N T_{jk}(t_{i+1}, t_i)p_k(t_i)$. Since this relationship can be applied recursively, if the state transition matrix is time invariant, then $p(t_i) = T^i p(t_0)$.

Now consider an M -step transition probability $T_{jk}(t_{i+M}, t_i)$. Let (t_{i+1}) be some intermediate time between (t_i) and (t_{i+M}) so that $1 < M$. Then

$$T_{jk}(t_{i+M}, t_i) = \sum_{l=1}^N T_{jl}(t_{i+M}, t_{i+1})T_{lk}(t_{i+1}, t_i)$$

Therefore, the conditional probability that the system will be in state j at time given that it is in state k at time (t_i) is equal to the summation (over the intermediate state l) of the N possible terms formed as the product of the transition (probability) from k to l and the transition (probability) from l to j . Since this is the jk^{th} element of $T(t_{i+M}, t_i) = T(t_{i+M}, t_{i+1})T(t_{i+1}, t_i)$, and is the discrete time version of the Chapman-Kolmogorov equation.

5.4.5. Chapman-Kolmogorov Equation

Note that the state transition probability matrix has the semigroup property associated with linear deterministic state models of dynamic systems: transitioning from (t_i) to (t_{i+M}) can be achieved as a transition from (t_i) to (t_{i+1}) , and then from there to (t_{i+M}) .

In the continuous-state case where the probability density functions are continuous, assume that the conditional density function exists:

$$f_{x(t)|x(t')}(x|\rho) = \frac{\partial}{\partial \xi_1 \dots \partial \xi_n} F_{x(t)|x(t')}(x|\rho)$$

With the notation:

$$f_{x(t)|x(t')}(x|\rho) \equiv f_x(x, t|x(t') = \rho) \equiv f_x(x, t|\rho, t')$$

we can write (from the definition of conditional densities and Bayes rule):

$$f_{x(t_3), x(t_2)|x(t_1)}(x, \rho|\eta) = f_{x(t_3)|x(t_2), x(t_1)}(x|\rho, \eta)f_{x(t_2)|x(t_1)}(x|\rho, \eta)$$

Assuming a Markov process:

$$f_{x(t_3), x(t_2)|x(t_1)}(x, \rho|\eta) = f_{x(t_3)|x(t_2)}(x|\rho)f_{x(t_2)|x(t_1)}(\rho|\eta)$$

The conditional marginal density for $x(t_3)$, given that $x(t_1) = \eta$, can be obtained by integrating over the process value ρ , the process value at the intermediate time (t_2):

$$\begin{aligned} f_{x(t_3)|x(t_1)}(\xi|\eta) &= \int_{-\infty}^{\infty} f_{x(t_3)|x(t_2),x(t_1)}(\xi,\rho|\eta) d\rho \\ &= \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} f_{x(t_3)|x(t_2)}(\xi|\rho) f_{x(t_2)|x(t_1)} d\rho_1 \cdots d\rho_2 \end{aligned}$$

This is the conventional form of the Chapman-Kolmogorov equation:

$$f_x(\xi, t_3 | \eta, t_1) = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} f_x(\xi, t_3 | \rho, t_2) f_x(\xi, t_2 | \eta, t_1) d\rho_1 \cdots d\rho_2$$

5.5. Nonlinear Ito Stochastic System Models

5.5.1. Ito Stochastic Integrals and Differentials

Consider an extension to $I(t, \cdot) = \int_{t_0}^t a(\tau, \cdot) d\beta(\tau, \cdot)$ where $a(\tau, \cdot)$ is an admissible stochastic process such that $a(\tau, \cdot)$ depends at most on the past and present values of $\beta(\tau, \cdot)$ but is independent of future values of $\beta(\tau, \cdot)$. Suppose that $\int_{t_0}^t E\{a(\tau, \cdot)^2\} d\tau$ is finite with probability one. Given these sufficient conditions the Ito stochastic integral can be defined as the mean square limit:

$$I(t, \cdot) = \int_{t_0}^t a(\tau, \cdot) d\beta(\tau, \cdot) = \text{l.i.m.}_{N \rightarrow \infty} \sum_{i=0}^{N-1} a_N(t_i, \cdot) [\beta(t_{i+1}, \cdot) - \beta(t_i, \cdot)]$$

Any finite-variance nonlinear functional of Brownian motion can be expressed as an Ito stochastic integral. Note: The Wiener integral can be considered as a special case of the Ito stochastic integral with a nonrandom function $A(t) = a(t)$.

5.5.1.1. Properties of Ito Stochastic Integrals

Viewed as a stochastic process $I(t, \cdot)$ is itself admissible, mean square continuous, continuous with probability one, and is a martingale of Brownian motion:

$$E\{I(t) | \{\beta(\tau), t_0 \leq \tau \leq t' \leq t\}\} = I(t')$$

However, over the interval (without the conditional expectation) $E\left\{\int_{t_0}^t a(\tau) d\beta(\tau)\right\} = 0$

$$\text{and } E\left\{\left[\int_{t_0}^t a(\tau) d\beta(\tau)\right]\left[\int_{t_0}^t b(\tau) d\beta(\tau)\right]\right\} = \int_{t_0}^t E\{a(\tau)b(\tau)\} d\tau$$

5.5.2. Nonlinear Ito Stochastic Differential Equations

Consider a dynamical system described by the nonlinear Ito stochastic differential equation:

$$dx(t) = f[x(t), t]dt + G[x(t), t]d\beta(t)$$

where $x(\cdot, \cdot)$ is an n -dimensional state stochastic process, $f[x(t), t]$ is an n -vector function describing system dynamics, $G[x(t), t]$ is an n -by- s matrix of functions, and $\beta(\cdot, \cdot)$ is an s -vector dimensional Brownian motion of mean zero and diffusion $Q(t)$:

$$E\{d\beta(t)d\beta^T(t)\} = Q(t)dt$$

$$E\{[\beta(t_2) - \beta(t_1)][\beta(t_2) - \beta(t_1)]^T\} = \int_{t_0}^{t_2} Q(t)dt$$

The dynamical system can be interpreted as

$$x(t) - x(t_0) = \int_{t_0}^t f[x(\tau), \tau]d\tau + \int_{t_0}^t G[x(\tau), \tau]d\beta(\tau)$$

with $G[\cdot, t]$ a function only of $x(t)$ rather than the entire history of $\{x(\tau), t_0 \leq \tau \leq t\}$. If the functions $f[\cdot, t]$ and $G[\cdot, t]$ were any admissible function, the solution would yield processes known as Ito processes. The restriction that $G[\cdot, t]$ be a function only of $x(t)$ only generates solutions that are Markov. It should be noted that only the Ito definition of the stochastic integral will lead to a Markov process.

Sufficient conditions for the existence of a unique solution:

- Both $f[x(t), t]$ and $G[x(t), t]$ are real functions that are uniformly Lipschitz in their first argument - continuity;
- There exists a K , independent of t , such that:
 $\|f[x + \Delta x, t] - f[x, t]\| \leq K\|\Delta x\|$ and $\|G[x + \Delta x, t] - G[x, t]\| \leq K\|\Delta x\|$ with appropriate norm definitions⁴;

$$\|f\| = \left[\sum_{i=1}^n v_i^2 \right]^{1/2} = [v^T v]^{1/2} = [\text{tr}(vv^T)]^{1/2}$$

4

$$\|M\| = \left[\sum_{i=1}^m \sum_{j=1}^n M_{ij}^2 \right]^{1/2} = [\text{tr}(MM^T)]^{1/2}$$

- Both $f[x(t), t]$ and $G[x(t), t]$ are continuous in their second (time) argument over the interval of interest;
- Both $f[x(t), t]$ and $G[x(t), t]$ are uniformly bounded according to $\|f[x, t]\|^2 \leq K(1 + \|x\|^2)$ and $\|G[x, t]\|^2 \leq K(1 + \|x\|^2)$;
- The process $x(\cdot, \cdot)$ is any random vector with finite second moment $E\{x(t_0)x^T(t_0)\}$ which is independent of the Brownian motion process $\beta(\cdot, \cdot)$.

5.5.3. Properties of the Solution Process

- The process $x(\cdot, \cdot)$ is continuous with probability one and is also mean square continuous such that:

$$\lim_{t' \rightarrow t} x(t') = x(t)$$

or

$$\lim_{t' \rightarrow t} \text{tr } E\{[x(t') - x(t)][x(t') - x(t)]^T\} = 0;$$

- Both $x(t)$ and $[x(t) - x(t_0)]$ are independent of future increments of $\beta(\cdot, \cdot)$;
- The process $x(\cdot, \cdot)$ is Markov. Thus the conditional probability distribution for $x(t)$ given $x(t')$ equals the distribution conditioned only on $x(t')$;
- The mean square value of each component of $x(\cdot, \cdot)$ is bounded by a finite number $E\{x_i(t)^2\} < M < \infty$ and $\int_{t_0}^t E\{x_i(t)^2\} dt < \infty$;
- The probability of a change in $x(t)$ in a small interval Δt is of higher order than Δt , therefore:

$$\lim_{\Delta t \rightarrow 0} \frac{1}{\Delta t} \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} f_x(\xi, t + \Delta t | \rho, t) d\xi_1 \dots d\xi_n = 0$$

$\|\xi - \rho\| \geq \delta$

where the integration is carried out outside a ball of radius δ ;

- The drift (rate of change in) of $x(\cdot, \cdot)$ (going from t to $t + \Delta t$ in the limit as $\Delta t \rightarrow 0$) is $f[x(t), t]$;

- The diffusion (covariance of the rate of change) of $x(\cdot, \cdot)$ at time t is $\{G[x(t), t]Q(t)G^T[x(t), t]\}$;
- The higher order infinitesimals in the probability of change, drift, and diffusion are all zero.

5.5.4. Ito Differential Rule

Formal rules of differentiation and integration are not valid for Ito stochastic integrals or differentials based on them. Differentials satisfy the Ito differential rule. Let $\psi[\cdot, \cdot]$ be a scalar real valued function that has continuous first and second partial derivatives with respect to its first argument and continuously differentiable in its second argument:

$$d\psi[x(t), t] = \frac{\partial \psi}{\partial t} dt + \frac{\partial \psi}{\partial x} dx + \frac{1}{2} \text{tr} \left\{ G[x(t), t] Q(t) G^T[x(t), t] \frac{\partial^2 \psi}{\partial x^2} \right\} dt$$

where

$$\frac{\partial \psi}{\partial t} \equiv \frac{\partial \psi[x, t]}{\partial t} \Big|_{x=x(t)}, \quad \frac{\partial \psi}{\partial x} \equiv \left[\frac{\partial \psi}{\partial x_1} \dots \frac{\partial \psi}{\partial x_n} \right]_{x=x(t)}, \quad \text{and} \quad \frac{\partial^2 \psi}{\partial x^2} \equiv \begin{bmatrix} \frac{\partial^2 \psi}{\partial x_1^2} & \dots & \frac{\partial^2 \psi}{\partial x_1 \partial x_n} \\ \vdots & & \vdots \\ \frac{\partial^2 \psi}{\partial x_n \partial x_1} & \dots & \frac{\partial^2 \psi}{\partial x_n^2} \end{bmatrix}_{x=x(t)}$$

The Ito differential rule can be combined with the Ito stochastic differential equation to write the differential rule in terms of a differential generator $\mathcal{L}\{\psi[x(t), t]\}$:

$$d\psi[x(t), t] = \frac{\partial \psi}{\partial t} dt + \mathcal{L}\{\psi[x(t), t]\} dt + \frac{\partial \psi}{\partial x} G[x(t), t] d\beta(t)$$

$$\mathcal{L}\{\psi[x(t), t]\} = \frac{\partial \psi}{\partial x} f[x(t), t] + \frac{1}{2} \text{tr} \left\{ G[x(t), t] Q(t) G^T[x(t), t] \frac{\partial^2 \psi}{\partial x^2} \right\} dt$$

5.5.5. Transition Probability Density

The transition probability density for $x(\cdot, \cdot)$, $f_x(\xi, t | \rho, t')$, satisfies the forward Kolmogorov equation:

$$\begin{aligned} \frac{\partial f_x(\xi, t | \rho, t')}{\partial t} = & - \sum_{i=1}^n \frac{\partial}{\partial \xi_i} \{ f_x(\xi, t | \rho, t') f_i[\xi, t] \} \\ & + \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \frac{\partial^2}{\partial \xi_i \partial \xi_j} \left[f_x(\xi, t | \rho, t') \{ G[\xi, t] Q(t) G^T[\xi, t] \}_{ij} \right] \end{aligned}$$

5.5.6. Propagation of the Mean and Covariance

Propagation of the mean and covariance is not feasible without knowledge of the entire density for all t . However, if $x(\cdot, \cdot)$ is Markov then its mean and covariance propagate according to:

$$\begin{aligned} \dot{m}_x(t) &= E\{f[x(t), t]\} \\ \dot{P}_x(t) &= [E\{f[x(t), t]x^T(t)\} - E\{f[x(t), t]\}m_x^T(t)] \\ &\quad + [E\{x(t)f^T[x(t), t]\} - m_x(t)E\{f^T[x(t), t]\}] \\ &\quad + E\{G[x(t), t]Q(t)G^T[x(t), t]\} \end{aligned}$$

The general solution to this set of equations, however, requires knowledge of the density to evaluate the expectations.

5.6. Ito Stochastic Prediction

5.6.1. Linear System Models

If the system model is linear: $dx(t) = F(t)x(t) + G(t)d\beta(t)$, solutions to the forward Kolmogorov equation can be obtained via characteristic functions. This yields the familiar form for the state and covariance update:

$$\begin{aligned} \dot{\hat{m}}_x(t) &= F(t)\hat{m}_x(t) \\ \dot{P}_x(t) &= F(t)P_x(t) + P_x(t)F^T(t) + G(t)Q(t)G^T(t) \end{aligned}$$

5.6.2. Nonlinear System Models

If the system dynamics are linear, or we are willing to neglect the second partial derivatives with respect to x , we can use the extended Kalman filter. Consider the general nonlinear model:

$$\dot{x}(t) = f[\hat{x}(t), u(t), t] + L(t)w(t)$$

with $x(t_0)$ modeled as a Gaussian random vector with mean \hat{x}_0 and covariance P_0 and a measurement model of:

$$z(t_i) = h[x(t_i), t_i] + v(t_i)$$

The **extended Kalman filter** for this model is the following set of equations:

State estimate extrapolation by integrating from time t_i to t_{i+1} :

$$\dot{\hat{x}}(t/t_i) = f[\hat{x}(t/t_i), u(t), t]$$

Error covariance extrapolation by integrating from time t_i to t_{i+1} :

$$\dot{P}(t/t_i) = F[t; \hat{x}(t/t_i)]P(t/t_i) + P(t/t_i)F^T[t; \hat{x}(t/t_i)] + G(t)Q(t)G^T(t)$$

Filter gain calculation with $\hat{x}(t_i^-) = \hat{x}(t_i / t_{i-1})$:

$$K(t_i) = P(t_i^-)H^T[t_i; \hat{x}(t_i^-)]\{H[t_i; \hat{x}(t_i^-)]P(t_i^-)H^T[t_i; \hat{x}(t_i^-)] + R(t_i)\}^{-1}$$

State estimate update:

$$\hat{x}(t_i^+) = \hat{x}(t_i^-) + K(t_i)[z(t_i) - h[\hat{x}(t_i^-), t_i]]$$

Error covariance update:

$$P(t_i^+) = \{I - K(t_i)H[t_i; \hat{x}(t_i^-)]\}P(t_i^-)\{I - K(t_i)H[t_i; \hat{x}(t_i^-)]\}^T + K(t_i)R(t_i)K^T(t_i)$$

where the following definitions apply:

$$F[t; \hat{x}(t/t_i)] \equiv \left. \frac{\partial f[x(t), u(t), t]}{\partial x} \right|_{x=\hat{x}(t/t_i)}$$

and

$$H[t; \hat{x}(t_i^-)] \equiv \left. \frac{\partial h[x, t_i]}{\partial x} \right|_{x=\hat{x}(t_i^-)}$$

If the nonlinear system is described by $dx(t) = f[x(t), t]dt + L[t]d\beta(t)$ where $L = L[t]$ and not $L[x(t), t]$, then the transition probability density propagates according to the following forward Kolmogorov equation:

$$\frac{\partial f_x(\xi, t)}{\partial t} = -\sum_{i=1}^n \frac{\partial}{\partial \xi_i} \{f_x f_i\} + \frac{1}{2} \text{tr} \left\{ \left[\frac{\partial^2 f_x}{\partial \xi_i \partial \xi_j} \right] \{G(t)Q(t)G^T(t)\} \right\}$$

In the general case, the nonlinear problem is not solvable. There are a number of other approximations that exploit a Taylor series representation of the dynamics and measurement to estimate conditional moments. One of the more computationally reasonable is the **modified Gaussian second order filter**. This filter accounts for fourth central moments and is given by the following equations:

Initial conditions:

$$\hat{x}(t_i / t_i) = \hat{x}(t_i^+)$$

$$P(t_i / t_i) = P(t_i^+)$$

Differential equations:

$$\dot{\hat{x}}(t_i / t_i) = f[\hat{x}(t_i / t_i), t] + b_p(t / t_i)$$

$$\dot{P}(t_i / t_i) = F[t; \hat{x}(t / t_i)]P(t / t_i) + P(t / t_i)F^T[t; \hat{x}(t / t_i)] + \overline{G[x(t), t]Q(t)G^T[x(t), t]}$$

where $F[t; \hat{x}(t / t_i)]$ is given by the n-by-n partial derivative matrix:

$$F[t; \hat{x}(t / t_i)] \equiv \left. \frac{\partial f[x, t]}{\partial x} \right|_{x=\hat{x}(t/t_i)}$$

and

$$b_{pk}(t / t_i) \equiv \frac{1}{2} \text{tr} \left\{ \frac{\partial^2 f_k[\hat{x}(t / t_i), t]}{\partial x^2} P(t / t_i) \right\}$$

This predictor follows the general structure of the Kalman filter. After integrating to the next sample time:

$$\hat{x}(t_{i+1}^-) = \hat{x}(t_{i+1} / t_i)$$

$$P(t_{i+1}^-) = P(t_{i+1} / t_i)$$

we can use the following for the measurement update at time t_i :

Gain calculation:

$$K_{GS}(t_i) = P(t_i^-)H^T[t_i; \hat{x}(t_i^-)]A_{GS}^{-1}(t_i^-)$$

State update:

$$\dot{\hat{x}}(t_i^+) = \hat{x}(t_i^-) + K_{GS}(t_i) \left\{ z_i - h[\hat{x}(t_i^-), t_i] \right\} + b_m(t_i^-)$$

Covariance update:

$$\dot{P}(t_i^+) = P(t_i^-) - K_{GS}(t_i) H[t_i; \hat{x}(t_i^-)] P(t_i^-)$$

with

$$A_{GS}(t_i) = -H[t_i; \hat{x}(t_i^-)] P(t_i^-) H^T[t_i; \hat{x}(t_i^-)] + \hat{B}_m(t_i^-) + R(t_i)$$

and the kl^{th} element of \hat{B}_m defined by:

$$\hat{B}_{mkl}(t_i^-) \equiv \frac{1}{2} \text{tr} \left\{ \frac{\partial^2 h_k[\hat{x}(t_i^-), t_i]}{\partial x^2} P(t_i^-) \frac{\partial^2 h_l[\hat{x}(t_i^-), t_i]}{\partial x^2} P(t_i^-) \right\}$$

6. REFERENCES

1. P. S. Maybeck, *Stochastic Models, Estimation, and Control, Vol 1*, Academic Press, 1982.
2. P. S. Maybeck, *Stochastic Models, Estimation, and Control, Vol 2*, Academic Press, 1982.
3. L. Ljung, *System Identification: Theory for the User*, Prentice-Hall, New Jersey, 1987.
4. Sinha, Kusta, *Modeling and Identification of Dynamic Systems*, Van Nostrand, 1983.
5. J. P. Norton, *An Introduction to and Identification*, Academic Press, 1988.
6. T. Kailath, *Linear Systems*, Prentice-Hall, New Jersey, 1980.
7. Franklin, Powel, Workman, *Digital Control of Dynamic Systems*, Addison Wesley, 1990.
7. H. Kwakernaak and R. Sivan, *Linear Optimal Control Systems*, Wiley-Interscience, 1972.
9. A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, McGraw-Hill, 1965.
10. A. Gelb, *Applied Optimal Estimation*, The M.I.T. Press, Cambridge, 1974.
11. L. Ingber, "Statistical Mechanics of Combat and Extensions," reprint from *Toward a Science of Command, Control, and Communications*, AIAA, December 1993.

CHAPTER 6

METAMODELING METHODS

1. CHAPTER OUTLINE

1. Chapter Outline	6-1
2. Introduction.....	6-3
2.1. Definitions and Notation	6-4
2.2. Historical Perspective.....	6-6
2.3. Revised Structure.....	6-8
2.3.1. Form of the Identifier.....	6-8
2.3.1.1. Equation Error	6-8
2.3.1.2. Output Error	6-9
2.3.1.3. Prediction Error	6-9
2.3.2. Criterion of Fit.....	6-10
2.3.2.1. Minimum Mean Square Error Estimators (MMSE).....	6-10
2.3.2.2. Maximum A Posteriori Probability (MAP) Estimators.....	6-11
2.3.2.3. Maximum Likelihood (ML) Estimators.....	6-11
2.4. Specific Identification Methods	6-12
2.4.1. Prediction Error and Correlation Approaches.....	6-12
2.4.2. Maximum Likelihood Approaches	6-13
2.4.3. Optimization.....	6-13
2.4.4. Approximation Techniques for Identification	6-14
3. Prediction Error Methods (PEM).....	6-15
3.1. General Description	6-15
3.2. Least Squares (LS).....	6-15
3.2.1. Ordinary Least Squares.....	6-16
3.2.1.1. Conditions for Existence of the Least Squares Solution	6-18
3.2.1.2. Statistical Properties of the Least Squares Estimator	6-19
3.2.2. Weighted Least Squares	6-19
3.2.2.1. Best Linear Unbiased Estimator (Gauss-Markov Estimate) for a Stochastic Model.....	6-20
3.2.2.2. Statistical Properties of the Weighted Least Squares Estimator	6-20
3.2.3. Minimum Variance Estimator	6-20
3.2.4. Recursive Least Squares	6-21
3.2.4.1. Ordinary and Weighted Recursive Least Squares.....	6-21
3.2.4.2. Weighted Recursive Least Squares.....	6-22
3.2.5. Ridge Regression.....	6-22
3.2.6. Chi-Square Fitting	6-22
3.3. Eigenstructure Realization Algorithm.....	6-24
3.4. Autoregressive Time Series Models.....	6-27

4. Correlation Approaches	6-28
4.1. Instrumental-Variable (IV) Method	6-28
4.2. Recursive Instrumental-Variables (IV)	6-29
5. Maximum Likelihood Approaches	6-30
5.1. Full Scale Estimator	6-30
5.1.1. Theory	6-30
5.1.2. Algorithm	6-32
5.2. Modified Maximum Likelihood (MMLE)	6-37
5.2.1. Maximum Likelihood (ML) Estimation	6-37
5.2.2. Maximum A Posteriori (MAP) Estimation	6-38
5.2.3. Output Error Method	6-38
6. Optimization	6-39
6.1. Problem Definition	6-39
6.2. Iterative Optimization Methods	6-40
6.3. pEsT	6-41
6.4. Simulated Annealing	6-41
7. Approximation Techniques for Identification	6-43
7.1. Quasi-Linearization	6-43
7.2. Stochastic Approximation	6-43
7.3. Spline Approximation	6-43
7.4. Canonical Variate Analysis (CVA)	6-44
7.4.1. Introduction	6-44
7.4.2. CVA Algorithm	6-47
7.5. State Space Reconstruction	6-48
8. Relationships Between Methods	6-50
8.1. ARX Models	6-50
8.2. Maximum Likelihood Approaches	6-50
9. References	6-52

2. INTRODUCTION

As of this time, we have chosen the system type and class, and have selected a model structure that we will use for the identification (Chapter 5). We now discuss techniques for generating the estimate. Consequently, this chapter continues with the presentation of research to support Objective 2 and covers the many methods available to support decisions associated with **"Step 12: Fit the metamodel."**

The combination of model selection, error criterion, identification technique, and numerical methods leads to an overwhelming myriad of "identification methods." In an attempt to remain as general as possible, much of this detail relating to specific cases will not be presented here. Also, much of the literature defines the identification technique by the numerical method used to arrive at the solution. As stated in the introduction, we do not specifically address this issue.

This research combined techniques from both engineering (system identification) and mathematics (statistics). System identification is the derivation of mathematical model from observed data [1]. However, our use of system identification is different from the usual practice in engineering. In control engineering, the purpose of the identification is to provide a model of suitable fidelity such that the application of inputs (derived from this model) to the actual system will provide some desired outcome (stability, tracking performance, etc.). Our use of system identification is to develop models like those used in science or statistics. Statistics is the branch of scientific method that deals with the data obtained by measuring the properties of general populations. Our objective is closer to that of statistics. We want to develop models that describe the system (or population) as it exists.

There seem to be as many system identification methods as there are inverse problems. Many specific identification and statistical methods have been developed to accommodate the differences in model structures, data length, and measurement error statistics, etc. And, in researching the literature of either system identification or statistics, one often finds considerable discussion on particular methods with very little discussion on the relationship of these techniques to each other or to a general methodology. Since we are looking for a connection between Air Force metamodeling problems and identification techniques, we discuss these methods as elements of a more general structure.

2.1. Definitions and Notation

To help with the discussion in this chapter, the following definitions and notation are repeated here.

Table 6.2.1. Notation.

CATEGORY	ELEMENTS	SYMBOL	REMARKS
Observations		Z^N	Data available from the simulation Elements of the observation vector are z_1, z_2, \dots, z_N
	Inputs	$u(t)$	dimension $\Rightarrow r$
	Outputs	$y(t_i)$	dimension $\Rightarrow m$
Latent variables		$x(t)$	dimension $\Rightarrow n$
System disturbances		$w(t)$	Strength - $E\{w(t)w^T(t+\tau)\} = Q(t)\delta(\tau)$ Ito stochastic representation - $d\beta(t)$
Measurement noise		$v(t_i)$	Strength $E\{v(t_i)v^T(t_j)\} = R(t_i)\delta_{ij}$
			Correlated process and measurement noise - discrete system $E\{w_d(t_i)v^T(t_j)\} = S(t_i)\delta_{ij}$
Parameter vector		θ	
Error/Residual		ε	
Cost function		$J(\hat{\theta})$	
Finite dimensional representation	Scalar or MFD		$A(t)y(t) = \frac{B(q)}{F(q)}u(t) + \frac{C(q)}{D(q)}e(t)$ with $A(q) = 1 + a_1q^{-1} + \dots + a_nq^{-n}$
	Time invariant state space		Continuous time: $\dot{x} = Fx(t) + Gu(t) + Lw(t)$ $y(t) = H(\theta)x(t) + N(\theta)u(t) + v(t)$ Discrete time: $x_{t_{i+1}} = A(\theta)x(t_i) + B(\theta)u(t_i) + M(\theta)w_d(t_i)$ $y(t_i) = C(\theta)x(t_i) + D(\theta)u(t_i) + O(\theta)w_d(t_i) + v(t_i)$

Table 6.2.1. Notation (Cont.).

CATEGORY	ELEMENTS	SYMBOL	REMARKS
Finite dimensional representation	Time varying state space		Continuous time: $\dot{x} = F(t)x$ $y(t) = H(t)x(t) + v(t)$ Discrete time: $x_{t_{i+1}} = A(t_i, \theta)x(t_i) + B(t_i, \theta)u(t_i) + M(t_i, \theta)w_d(t_i)$ $y(t_i) = H(t_i, \theta)x(t_i) + v(t_i)$
	State matrix	$F(t)$	State transition matrix - $\Phi(t, t_0)$ Discrete system => $A(t) = \exp\{F(t - t_0)\}$
	Input matrix	$G(t)$	Discrete system => $B(\theta) = \int_{t_i}^{t_{i+1}} \Phi(t_{i+1}, \tau)G(\tau)d\tau$
	Observation matrix	$H(t_i)$	Discrete system => $C(\theta) = H(t_i)\Phi(t_i, t_i)$
	Disturbance matrix	$L(t)$	Discrete system => $M(\theta) = \int_{t_i}^{t_{i+1}} \Phi(t_{i+1}, \tau)L(\tau)d\tau$
	Nonlinear system		$\dot{x} = f(t, x(t), u(t), w(t); q)$ $y(t) = h(t, x(t), u(t), v(t); q)$

The identification method includes reference to observations, inputs, outputs, latent variables, parameter vectors, cost functions, errors, and residuals. Since the model structures outlined in Chapter 5 will define/limit the relationships that can be considered in the parameterization of the metamodel, this data is summarized again in Table 6.2.2.

Table 6.2.2. Model Structures and Limitations.

STRUCTURE	REMARKS
Autoregressive (AR)	Output is a function of the past outputs only
Autoregressive moving average (ARMA)	Output is a function of the past outputs and a moving average of the error (latent) variables
Moving average (MA)	Output is a function of latent variables only
Input/State/Output	The behavior satisfies the axiom of state

2.2. Historical Perspective

System identification techniques have been classified in a number of ways. This subsection gives some of the typical perspectives generally found in the literature. The focus is on particular techniques and how they can be applied.

Off-line methods often require the application of a special input and usually require storage of all of the data. They are run in what is called a "batch" (all at one time) mode. **On-line** methods do not require the application of a special input or storage of all data. On-Line algorithms are recursive and allow computation within one sampling interval. Also, these methods can be classified as either **open-loop** or **closed-loop** methods. An open-loop method requires that the system (also called the plant) to be identified be isolated from the environment and any feedback or control paths that modify the input. Closed-loop methods identify the system in its environment. Tables 6.2.3 and 6.2.4 present identification methods using this taxonomy.

Table 6.2.3. Open-Loop and Off-Line Identification Methods.

CLASSIFICATION	TECHNIQUE	REMARKS
Open Loop	Classical	Frequency response method Step response Impulse response Deconvolution: Determination of the impulse response from the input -- output map Correlation Random input
	Least squares	Weighted Sequential Generalized
Off-Line	Maximum likelihood approaches	Estimates of parameters from noise-contaminated data
	Prediction error method	
	Instrumental variable method	
	Stochastic modeling	Adaptive simulated annealing [2,3,4]

Further classification can be made as nonparametric, frequency domain, and as parameter identification methods. **Nonparametric** open-loop time domain methods include step response, impulse response, deconvolution, and correlation. **classical open-loop frequency domain** methods include frequency response analysis, fourier analysis, and spectral analysis.

Parameter identification methods are used when the candidate model is to be defined by a set of parameters. Parameter estimation algorithms mentioned in the literature include least squares, sequential weighted least squares, recursive generalized least squares, instrumental variables, recursive instrumental variables, the bootstrap method, sequential correlation, and recursive maximum likelihood estimation.

Table 6.2.4. On-Line and Closed-Loop Identification Methods.

CLASSIFICATION	TECHNIQUE	REMARKS
On-Line	Sequential weighted least squares	Models obtained using regression polynomial models Stationary time series models
	Recursive generalized least squares	First order autoregressive (AR1)
	Recursive instrumental variable method	First order moving average model (MA1)
	Bootstrap method	
	Sequential correlation method	nth- Order moving average model (MA _n) First order autoregressive moving average model (ARMA1,1)
	Recursive maximum likelihood estimation	First order autoregressive integrated moving average model (ARIMA1,1)
	Stochastic modeling	Nonstationary time series models
Closed-Loop systems	Learning model approach	
	Direct identification of systems w/ feedback	
	Estimation of forward path transfer function	
Continuous-Time systems	Direct method	
	Indirect method	

Since we are merely discussing different representations of the same system, parametric and non-parametric methods are closely related. Nonparametric methods, however, often require a special input to an open-loop system. In metamodeling combat simulations, this is not an available alternative. Consequently, we will concentrate on parametric methods.

2.3. Revised Structure

Most of the above techniques can be classified by two elements of the identification method: the form of the identifier and the criterion of fit. Since we do not know the values of the parameter vector θ_* , we cannot define a parameter error between $\hat{\theta}$ and θ_* . The error must be computed from $\{z(t_i)\} \Leftrightarrow \{u(t_i)\}$ and $\{y(t_i)\}$. The **form of the identifier** defines the "experimental setup" or the manner in which the estimates are generated and compared. The **criterion of fit** establishes both the cost function and the method of its minimization.

2.3.1. Form of the Identifier

2.3.1.1. Equation Error

For the equation error method, Figure 6.2.1, we need the system equations as given. Assume first that we have the following general description defined by a parameter vector θ and that we know the form of the vector functions f and h :

$$\begin{aligned}\dot{x} &= f(t, x(t), u(t), w(t); \theta) \\ y(t) &= h(t, x(t), u(t), v(t); \theta)\end{aligned}$$

Now we assume that we can measure the controls, the states, and the state derivatives. With all of this information, we can determine the error between the model and the actual data: \dot{x}_a, x_a, u_a :

$$\varepsilon(t, \theta) = \dot{x}_a - f(x_a, u_a; \theta)$$

The vector $\varepsilon(t, \theta)$ is the equation errors.

From these equation errors, $\varepsilon(t, \theta)$, we can form some nonnegative function such as $J(\theta) = \int_0^T \varepsilon^T(t, \theta) \varepsilon(t, \theta) dt$ and search over θ to find the minimum ($J_{\min}(\theta) = 0$ if there is no noise present).

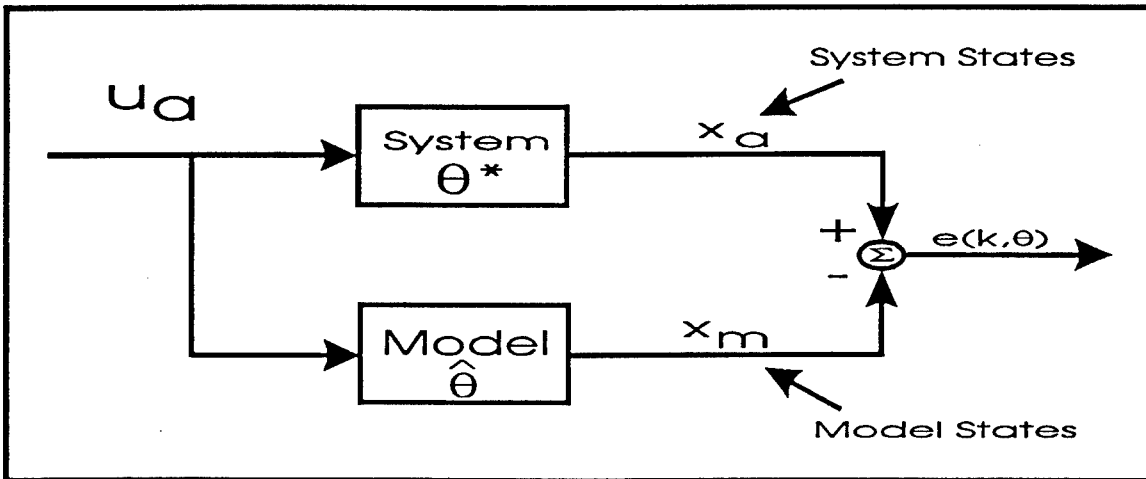


Figure 6.2.1. Experimental Format for the Equation Error Method.

2.3.1.2. Output Error

The general case of the equation error method required measurement of all of the elements of the system. Often, this is not possible. The output error method is based on an output error criterion and avoids this requirement.

Figure 6.2.2 depicts the experimental setup for the output error method. As you see, there is no attempt to measure the state of the plant. Instead, the estimated parameter, $\hat{\theta}$, is used in the model with the input u_a to generate an estimate of the output y_m . Again, we can form some nonnegative function of the difference between y_m and y_a . This time, however, the criterion function will include the model output in place of the model states.

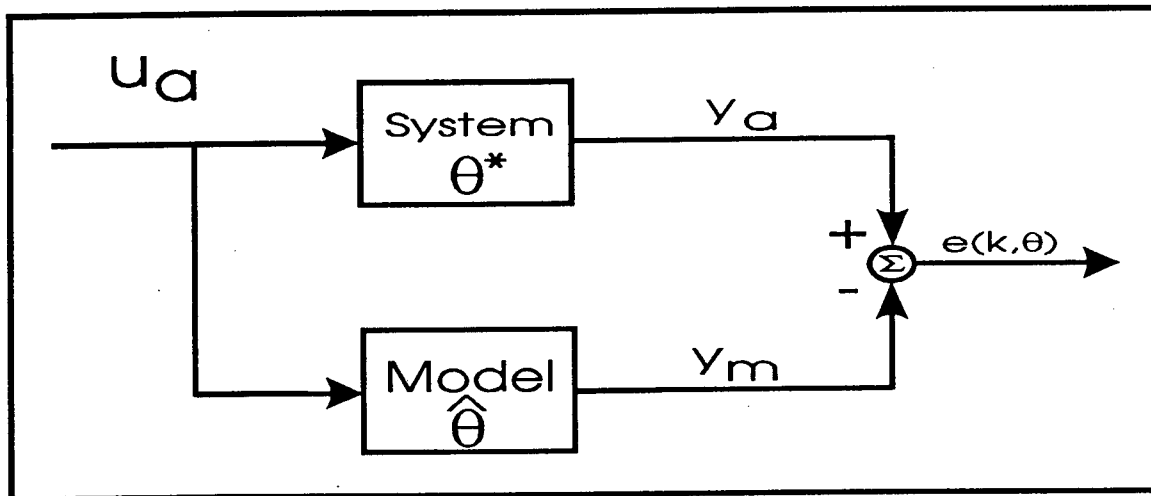


Figure 6.2.2. Experimental Format for the Output Error Method.

2.3.1.3. Prediction Error

The prediction error method is the third approach to developing an error function by which a parameter search can be structured.

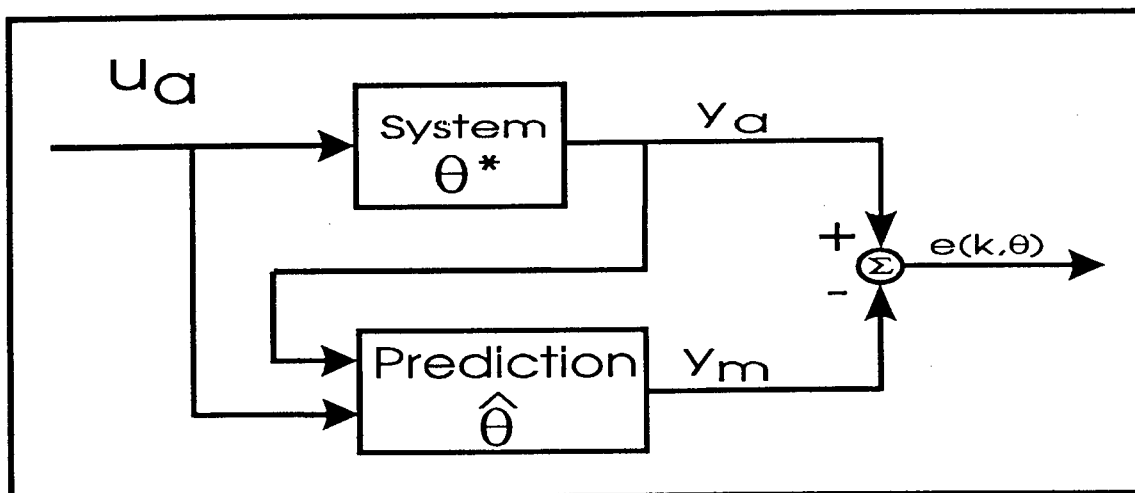


Figure 6.2.3. Experimental Format for the Prediction Error Method.

2.3.2. Criterion of Fit

This selection often defines the identification method. It is quite complicated and has many dimensions. Our framework is based on the assumption that the data will contain both measurement errors and system disturbances not accounted for by the model. Consequently, measurements are realizations of a stochastic system and are represented by functions of random variables that have some probability density function.

The probability that a particular random variable is in the range $a \leq x_i \leq b$ is given by:

$\Pr\{a \leq x_i \leq b\} = \int_a^b f_{x_i}(\xi) d\xi$ where $f_{x_i}(\xi)$ is the probability density function of the set of $\{x_i\}$. Therefore, the probability density function is a measure of the "likelihood" of a particular value.

Assume that the probability density function (PDF) of the measurements Z^N is $f(\theta; z_1, z_2, \dots, z_N) = f_z(\theta; Z^N)$ where θ is a d -dimensional parameter vector determined by the parameter estimator. This PDF is a joint PDF (JPDF) that considers the joint (combined) probability of both θ and Z^N occurring. However, because we have a function of a random variable and measurements that are available in a specific sequence, we can also consider the conditional probability distributions (CPDF). That is, the probability of an event conditioned on the fact that another event has occurred such as $P(z|\hat{\theta})$ which is the probability of a particular Z^N conditioned on the fact that $\theta = \hat{\theta}$.

By criterion of fit, we mean the function or functional that is optimized to determine the parameter estimates. It is entirely possible that the identification method, given a model and a particular set of data, has multiple characteristics. For example, least squares is a specific case of the prediction error method that minimizes a norm of the prediction error. Yet, if the data meets the assumptions of the method, least squares is also a maximum likelihood estimator since it also maximizes the likelihood of the parameter vector given the observations $f_z(\theta; Z^N)$. We consider three criterion: minimum mean square, maximum a posteriori (maximize the CPDF), and maximum likelihood (maximize the JPDF).

2.3.2.1. Minimum Mean Square Error Estimators (MMSE)

Minimum mean square estimators minimize a cost function that is a function of the (possibly weighted) output error only - $J(\hat{\theta}) = \epsilon^T W \epsilon$. It should be noted that the minimum mean square estimate (m.s.e.) is not necessarily the unbiased minimum variance estimator. The mean square error matrix M for an estimate of $\hat{\theta}$ of θ (with b equal to the bias) is:

$$M = E\left\{(\hat{\theta} - \theta)(\hat{\theta} - \theta)^T\right\} = \text{cov } \hat{\theta} + b b^T$$

Both bias and covariance must be minimized to attain the minimum mean square estimate; and, in general, the minimum m.s.e. will be biased. The minimum m.s.e. estimator will, however, result in output errors (residuals) that are orthogonal to the estimate.

2.3.2.2. Maximum A Posteriori Probability (MAP) Estimators

The Bayesian approach to parameter estimation assumes a parameter vector with *a priori* (before the measurement) probability densities $P(\theta)$. The observations Z^N , therefore, are correlated with θ . Measurements are used to determine the most likely value after the measurement, - Maximum a posteriori (MAP) estimate $\hat{\theta}_{MAP} = \arg(\hat{\theta}) \max_{\theta} P(\hat{\theta} | z)$ - via the application of Bayes rule:

$$P(\hat{\theta} | z) = \frac{P(z | \hat{\theta}) \times P(\hat{\theta})}{P(z)}$$

Here $P(z | \hat{\theta})$ is the conditional probability; i.e., the total probability of the measurement conditioned on the current estimate of θ .

We can rewrite the maximization to be the minimization of the negative logarithm of $P(\hat{\theta} | z)$:

$$\hat{\theta}_{MAP} = \arg(\hat{\theta}) \min_{\theta} [-\log P(\hat{\theta} | z)]$$

where $\log P(\hat{\theta} | z) = \log P(z | \hat{\theta}) + \log P(\hat{\theta}) - \log P(z)$. Since $P(z)$ is unaffected by $\hat{\theta}$, it can be ignored in the minimization.

Assuming the correct *a priori* probability, the MAP estimate minimizes the $E\{(\hat{\theta} - \theta)(\hat{\theta} - \theta)^T\}$ and, therefore, is the minimum-quadratic-cost estimate. The MAP estimate also minimizes the expected absolute error $E\{\|\hat{\theta} - \theta\| | Z^N\}$.

2.3.2.3. Maximum Likelihood (ML) Estimators

Given that the joint probability of the random vector to be observed is $f_z(\theta ; Z^N)$, then the probability that the random variable will produce the realization Z^N is proportional to $f_z(\theta ; Z^N)$. Once a particular realization Z^N is inserted into the joint PDF, this becomes deterministic and is called the *likelihood function*. A maximum likelihood estimator maximizes this function: $\hat{\theta}_{ML} = \arg(\theta) \max_{\theta} f_y(\theta ; Z^N)$ so that the observed event becomes as likely as possible.

Beginning with the MAP estimate and ignoring the prior information, we have for the ML estimate $\hat{\theta}_{ML} = \arg(\hat{\theta}) \min_{\theta} [-\log P(z | \hat{\theta})]$.

Since for $N = i$ we have $P(z|\hat{\theta}) = P(z_i|z_{i-1}, \dots, z_1, \hat{\theta}) \times P(z_{i-1}|z_{i-2}, \dots, z_1, \hat{\theta}) \times \dots \times P(z_1|\hat{\theta})$, we can define $-\log P(z|\hat{\theta}) = -\sum_{i=1}^N \log P(z_i|z_{1:i-1}, \hat{\theta})$ as the log likelihood function (LLF). Comparing the ML and MAP LLFs, we see that $LLF_{MAP} = LLF_{ML} + \log P(\hat{\theta})$.

Statistical properties of maximum likelihood estimators for "sufficiently long" data [5]:

1. Parameter errors have an unbiased Gaussian distribution.
2. Estimates are consistent (unbiased as the data length increases).
3. Efficient estimates; no unbiased estimator has lower error variance.

However, the MLE has been criticized for poor small sample properties.

2.4. Specific Identification Methods

Assume that a model structure (set of candidate models) has been selected and parameterized using some parameter vector θ . We have defined the model class $\mathcal{M}(\theta)$. The next step is to search for the best model within the set (determine the parameter vector θ). Recall that the objective is to determine the most powerful unfalsified model (MPUM) where a model is the MPUM based on the data \mathbf{D} if: (1) $M \in \mathcal{M}$; (2) \mathcal{M} is unfalsified by \mathbf{D} ; and (3) M is more powerful than any other model satisfying (1) and (2). We must determine the mapping from the data set \mathbf{D} to M .

2.4.1. Prediction Error and Correlation Approaches

Let the prediction error be given by $\varepsilon(t, \theta) = y(t) - \hat{y}(t|\theta)$. A "good" model will have small prediction errors. There are two general approaches in defining a measure of ε . The first is to define a norm that measures the size of ε and minimize that norm. This leads to the **prediction error method** (PEM). Another approach to define a measure of ε is to require that ε be uncorrelated with past data. This **correlation approach** contains the instrumental-variable (IV) method [6].

In addition to least squares (LS), subsets of the prediction error method also include the maximum likelihood approaches (ML and MAP). Our discussion, however, separates out the maximum likelihood approaches from PEM. We do so because when we consider probabilistic models (where ML and MAP estimators apply), the prediction equations for explicitly using the PEM algorithm are limited to the directly parameterized form. From Chapter 5, we see that there are a number of other probabilistic model structures where the PEM algorithm cannot be used.

We do include, however, the **eigenstructure realization algorithm** (ERA) under the section on PEM. We do so because this algorithm uses the least squares approach to directly identify the Markov parameters of a steady state Kalman filter.

We do include, however, the **eigenstructure realization algorithm (ERA)** under the section on PEM. We do so because this algorithm uses the least squares approach to directly identify the Markov parameters of a steady state Kalman filter.

2.4.2. Maximum Likelihood Approaches

If we consider independent, identically distributed measurements, and if an efficient estimate (unbiased estimate with finite covariance such that no other unbiased estimate has a lower covariance) exists it can always be found through maximum likelihood approaches. Again, if an efficient estimate exists, the likelihood equation will have a unique solution that equals the efficient estimate. If any single sufficient statistic exists, the maximum likelihood estimate will be sufficient. Although the maximum likelihood estimate will be biased for small samples, it will provide the unique minimum variance estimate attaining the Cramér-Rao lower bound if this is possible [7].

The objective is to provide a parameter estimator that does not require complete *a priori* parameter statistics yet still allows the inclusion of a priori knowledge. Unlike the best linear unbiased estimate provided by appropriately weighted least squares, this method propagates the probabilistic information in time and directly allows the inclusion of known statistical information.

These approaches can be used both on and off line. On-line identification systems are used in "adaptive" or "self tuning" that combine state and parameter estimation. Off-line parameter estimators do not need to consider the state estimation problem.

The key to the identification algorithm will be the residuals of the state estimator, and the most significant drawback of the maximum likelihood approaches is the lack of theoretical knowledge on the behavior of the estimates for small sample sizes.

The methods discussed below will assume that the parameters are constant. Time varying systems can be handled by repeating the identification every N data points.

2.4.3. Optimization

Often we are unable to formulate the problem such that a suitable prediction equation is available. Therefore we must resort to either a "nonlinear state space model" or a "simulation model." In these situations, where we are unable or unwilling to consider a linearized or perturbation approach, the best we can do is take the output of the model, incorporate it into a "cost" function, and adjust the model parameters to optimize (minimize) that function.

There are several "standard" numerical procedures that are used to search for the minimum of a function. These are outlined in **iterative optimization methods**. In addition, there are several programs that are designed to perform parameter estimation. An interactive program for parameter estimation of nonlinear dynamic systems, **pEst** uses three separate minimization algorithms (steepest descent, modified Newton-Raphson, and Davidon-Fletcher-Power) to minimize the following cost function:

$$J(\hat{\theta}) = \frac{1}{2n_N n_z} \sum_{i=1}^{n_i} [z(t_i) - \hat{z}(t_i)]^T W [z(t_i) - \hat{z}(t_i)]$$

where n_N = number of data points and n_z = number of response variables.

Using statistical mechanical theories, an optimization technique called "**simulated annealing**" provides a new option to directly process nonlinear, discontinuous, stochastic functions [2]. Given data and a cost function, it will globally optimize that function. Simulated annealing is a form of the Metropolis algorithm. Given a description of possible system configurations and an objective function to minimize, this technique emulates physical annealing to arrive at a global minimum.

2.4.4. Approximation Techniques for Identification

The first approximation technique we mention is **quasi-linearization**. The only reason we mention the technique is for completeness, since no additional algorithms are actually introduced.

Stochastic approximation may be regarded as the application of gradient methods to stochastic problems. It is a scheme for successive approximation of a sought quantity when the observations involve random errors due to the stochastic nature of the problem. The main advantage is the simplicity of the implementation and the fact that prior knowledge of the noise statistics are not necessary.

Polynomials are excellent approximating functions when a smooth function is to be approximated locally. Any such smooth piecewise polynomial function is called a **spline**, and they are commonly used for fitting data.

Another approximation technique is **canonical variate analysis**. The canonical variate method is a prediction error approximation technique that optimally predicts future responses based on a reduced order state space system. Derived by considering the fact that the conditional expectation is an optimal projection in Hilbert space, the procedure optimally selects k linear combinations of the past data for prediction of the future.

Our final approximation technique, **state space reconstruction** generates a state space model from an optimal prediction of the future states from linear combinations of the past. Given the data from CVA, or any other identification method, we can use these predictions to parameterize a state space system for any order $k < q$ via a least squares regression.

3. PREDICTION ERROR METHODS (PEM)

3.1. General Description

Filter the prediction sequence $\varepsilon(t, \theta)$ using a stable linear filter $L(q)$:

$$\varepsilon_F(t, \theta) = L(q)\varepsilon(t, \theta) \quad 1 \leq t \leq N$$

This filtering acts like frequency weighting and can remove or enhance selected properties of the model. (In linear systems this is equivalent to prefiltering the data by $L(q)$). Then, using either a fixed or weighted (possibly time varying) norm:

$$V_N(\theta, D) = \frac{1}{N} \sum_{t=1}^N \ell(\varepsilon_F(t, \theta), \theta, t)$$

or

$$V_N(\theta, D) = \sum_{t=1}^N \beta(N, t) \ell(\varepsilon_F(t, \theta), \theta, t)$$

the estimate $\hat{\theta}_N$ is defined by the minimization:

$$\hat{\theta}_N = \hat{\theta}_N(D) = \arg \min_{\theta \in D} \{V(\theta, D)\}$$

Specific methods are obtained as special cases of the PEM with special selections of the filter $L(q)$ and the scalar valued norm function $\ell(\cdot)$. In general, the PEM is a technique of approximating (smoothing) the empirical transfer function estimate to the model transfer function with a weighted norm corresponding to the model signal to noise at the frequency in question.

3.2. Least Squares (LS)

If the predictor is linear, the prediction error becomes $\varepsilon(t, \theta) = y(t) - \varphi^T(t)\theta$ where $\varphi(t)$ is the vector of regressors that depends on the selected model structure. Also if $L(q) = 1$ and $\ell(\varepsilon) = \frac{1}{2} \varepsilon^2$, then the norm becomes:

$$V_N(\theta, D) = \frac{1}{N} \sum_{t=1}^N \frac{1}{2} [y(t) - \varphi^T(t)\theta]^2$$

This is the least squares criterion for linear regression. The linear parameterization and quadratic criterion results in a quadratic function in θ that can be minimized analytically [6].

While we have discussed least squares as a parameter estimation method of system identification, linear regressions are probably the most common method of statistics; the branch of scientific method that deals with the data obtained by measuring the properties of general populations [8]. Because of the widespread use, many desirable properties, and the importance of this method, we will discuss it further.

Least squares, however, does come with a number of very limiting assumptions. (Most of these are due to the lack of dynamics or latent variables in the identification process.) It should also be noted that the model is a function of current inputs only and it is not a function of past inputs or past output values, and the least squares can never truly duplicate saturation, or asymptotic behavior.

The model for least squares requires statistically independent measurements with a common variance[9]. If the variance of the output is not the same for all measurements or if there is correlation between the measurements, a transformation of coordinates is required. Another basic assumption of the least squares regression technique is homoscedasticity: the deviation of the error terms is constant and the error is independent of the magnitude of the variables. Because of these assumptions, there may be something basically unrealistic in the use of multiple linear regression to describe highly complex, nonlinear systems.

Least squares fitting is maximum likelihood estimation if the measurement errors are independent and normally distributed with constant standard deviation.[10]

3.2.1. Ordinary Least Squares

Assume the following model¹:

$$y = X\theta + \varepsilon$$

where:

y	is an	$n \times 1$	vector of responses
X	is an	$n \times p$	matrix of inputs (basis functions)
θ	is an	$p \times 1$	vector of metamodel coefficients
ε	is an	$n \times 1$	vector of error terms

¹As indicated earlier, this is a fairly general structure. For example, an n^{th} order ARX model can be placed in this form by defining the vector of responses $y = [y(k) \ y(k-1) \cdots y(k-n)]^T$ and, with $n+m=p$, $\phi^T(k) = [-y(k-1) \cdots y(k-n) \ u(k-1) \cdots u(k-m)]$ and the basis functions as $X = [-\phi(k) \ \phi(k-1) \cdots \phi(k-n)]^T$

Also, assume that the variance for all measurements is constant and equal to σ^2 . A further refinement, chi-squared fitting (see below), normalizes the design matrix and output vector by the variance in the measurements.

The estimated model is $\hat{y} = X\theta$ with an error of $\varepsilon = \hat{y} - y$. If we define the norm squared error $J = \varepsilon^T \varepsilon$, we have:

$$\begin{aligned} J &= \varepsilon^T \varepsilon \\ &= (y - X\theta)^T (y - X\theta) \\ &= y^T y - y^T X\theta - \theta^T X^T y + \theta^T X^T X\theta \end{aligned}$$

Minimizing $J = \varepsilon^T \varepsilon$, we have the least squares estimator:

$$\begin{aligned} \frac{\partial}{\partial \theta} [J] &= 0 \\ \frac{\partial}{\partial \theta} [(y - X\theta)^T (y - X\theta)] &= 0 \\ -2y^T X + 2X^T X\theta &= 0 \\ \theta &= (X^T X)^{-1} X^T y \end{aligned}$$

The sum of squares are an indication of the variation of the model. Defining terms for multiple linear regression:

If we define the components of X as x_{ij} , the corrected sum of squares for the i th regressor variable becomes:

$$S_{ii} = \sum_{j=1}^n (x_{ij} - \bar{x}_i)^2 = \sum_{j=1}^n x_{ij}^2 - \frac{(\sum_{j=1}^n x_{ij})^2}{n}$$

The corrected sum of cross products of inputs r and s (x_r and x_s for $r \neq s$):

$$S_{rs} = \sum_{j=1}^n (x_{rj} - \bar{x}_r)(x_{sj} - \bar{x}_s) = \sum_{j=1}^n x_{rj} x_{sj} - \frac{(\sum_{j=1}^n x_{rj})(\sum_{j=1}^n x_{sj})}{n}$$

The corrected sum of cross products of x_i and y :

$$S_{iy} = \sum_{j=1}^n y_j (x_{ij} - \bar{x}_i) = \sum_{j=1}^n x_{ij} y_j - \frac{(\sum_{j=1}^n x_{ij})(\sum_{j=1}^n y_j)}{n}$$

The total sum of squares (S_{yy}) can be divided into two parts, the regression sum of squares (SS_R) and the residual or error sum of squares (SS_E). The residuals sum of squares is variation left unexplained:

$$\begin{aligned} SS_E &= \sum_{j=1}^n e_j^2 = \sum_{j=1}^n (y_j - \bar{y}_j)^2 \\ &= y'y - \theta' X'y \\ &= S_{yy} - \sum_{j=1}^k \theta_j S_{iy} \end{aligned}$$

Since $S_{yy} = SS_E + SS_R$, the variation explained by the model, the regression sum of squares is $SS_R = \sum_{j=1}^k \theta_j S_{iy} = \theta' X'y$

The mean square error (estimate of σ^2) of the estimate - sometimes represented by s^2 to preclude confusion with an assumed or "known" σ^2 :

$$\begin{aligned} MS_E &= \frac{y'y - \theta' X'y}{n - p} \\ &= \frac{\sum_{j=1}^n y_j^2 - \theta' X'y}{n - p} \end{aligned}$$

The least squares problem suffers from being both over determined (number of data points greater than the number of parameters) and underdetermined (linear combinations of input data). The Singular Value Decomposition (SVD) is designed to handle both overdetermined and underdetermined systems and offers a more stable numerical solution.

The SVD of X results in the generation of three matrices such that $X = U \cdot S \cdot V^T$. U and V are unitary matrices, while S is a diagonal matrix. With these matrices (with $U(i)$ and $V(i)$ defined as the columns of the matrices and $S(i)$ the singular values) the solution to the normal equations becomes:

$$\theta = \sum_{i=1}^N \left(\frac{U(i) \cdot y}{S(i)} \right) V(i)$$

3.2.1.1. Conditions for Existence of the Least Squares Solution

Even though the least squares estimate can be calculated without the inversion of $(X^T X)$ via a singular value decomposition, the matrix $(X^T X)$ still must be of full rank for the estimate to exist. This means that input sequence must be persistently exciting.

3.2.1.2. Statistical Properties of the Least Squares Estimator

1. For the deterministic model $y = X\theta + \varepsilon$, $\hat{\theta}$ is an unbiased estimator

$$\begin{aligned} E\{\theta\} &= E\{(X'X)^{-1}X'y\} \\ &= E\{(X'X)^{-1}X'(X\theta + \varepsilon)\} \\ &= E\{(X'X)^{-1}X'XX'\theta + (X'X)^{-1}X'\varepsilon\} \\ &= \theta \end{aligned}$$

2. If measurement errors are present, $\hat{\theta}$ can be a consistent estimator (see next section). In this case, however, the error terms are random variables that are a function of both the measurement noise and the choice of $\hat{\theta}$. The estimation error will be correlated with X , and the estimate will not be unbiased. Consequently, if the model includes measurement noise, the estimates will be biased. Consider the situation where the process model is $y_0 = X_0\theta + \varepsilon$, the output measurement error is $y = y_0 + V$ and the input measurements themselves have errors $X = X_0 + U$:

$$E\{\theta\} = -E\left\{\left[(X_0 + U)^T(X_0 + U)\right]^{-1}[X_0 + U]^T U\right\}\theta$$

We see that the bias is due solely to the correlation between modeling error and input measurement error only. Also, an unbiased estimate can be derived from the weighted least squares (with a specific weight) or the instrumental variable methods shown below.

3. Covariance.

Since we have assumed that $Var(\varepsilon_i) = \sigma^2$, it follows that $Var(y_i) = \sigma^2$, and

$$\begin{aligned} Cov(\theta) &\equiv E\{[\theta - E\{\theta\}][\theta - E\{\theta\}]^T\} \\ &= \sigma^2(X'X)^{-1} \end{aligned}$$

Therefore the standard deviation of the coefficients is $\sqrt{Cov(\theta)} = \sigma\sqrt{(X'X)^{-1}}$, but this has real meaning only to the extent that the measurement errors are normally distributed.

From the solution using the SVD, the covariance matrix can be computed by:

$$Cov(\theta_i, \theta_j) = \sum_{k=1}^N \left(\frac{V_{ik} V_{jk}}{S_k^2} \right)$$

3.2.2. Weighted Least Squares

The performance measure $J = \varepsilon^T \varepsilon$, was based on the view that all errors are equally important. **Weighted least squares** weights the errors and is based on the criterion $J = \varepsilon^T W \varepsilon$. With this error criterion, the normal equations for weighted least squares become: $\theta_{WLS} = (X^T W X)^{-1} X^T W y$. Many weighting functions are available. The weight,

$w = a\gamma^{n-k}$, weights the most recent observations more than the past and corresponds to a first order filtering of the data. As a approaches zero, the memory becomes long; whereas as a approaches 1 the memory becomes short. The selection of $w = a\gamma^{N-k}$, where N is the number of equations, and k is the column of X , provides the ability to select between ordinary least squares ($a = \gamma = 1$) and exponentially weighted least squares ($a = 1 - \gamma$ with $0 < \gamma < 1$).

3.2.2.1. Best Linear Unbiased Estimator (Gauss-Markov Estimate) for a Stochastic Model

As stated above, when measurement errors are present, the proper weights and the weighted least squares method can be used to give the **best linear unbiased estimator** (BLUE) or the Gauss-Markov estimate [11].

The best linear unbiased estimator is derived from the **stochastic model** $Y = X\theta_0 + V$ with the parameter estimate of the form $\hat{\theta} = LY$. We assume the covariance of the noise to be $E\{VV^T\} = R$ and require that the parameter estimate be (1) unbiased such that $E\{\hat{\theta}\} = \theta_0$; and (2) that it minimize the mean square **parameter** error:

$$J(L) = \text{tr} \left\{ E \left\{ (\hat{\theta} - \theta_0)(\hat{\theta} - \theta_0)^T \right\} \right\}$$

This formulation (via Lagrange multipliers) leads to $\hat{\theta}_{\text{BLUE}} = \hat{L}Y = (X^T R^{-1} X)^{-1} X^T R^{-1} Y$ which is a weighted least squares with the weight equal to the inverse of the measurement noise $W = R^{-1}$.

3.2.2.2. Statistical Properties of the Weighted Least Squares Estimator

For the stochastic least squares estimator to be consistent, the following must hold:

1. The measurement noise must be nonsingular. In this case, the input is said to be persistently exciting.
2. Either the noise must be a sequence of zero mean, independent random variables (white noise), or the input sequence must be independent of the zero mean noise sequence. These conditions will insure that $E\{X(t)v(t)\} = 0$.

3.2.3. Minimum Variance Estimator

If the variance of the parameters is known (or assumed), then we can further improve on the best linear unbiased (Gauss-Markov) estimator. This estimator is the minimum variance estimator. Assume that $E\{\theta\theta^T\} = Q$. Then the minimum variance estimator is:

$$\hat{\theta}_{\text{MV}} = (X^T R^{-1} X + Q^{-1})^{-1} X^T R^{-1} Y$$

and the corresponding error covariance is:

$$E\left\{\left(\hat{\theta} - \theta_0\right)\left(\hat{\theta} - \theta_0\right)^T\right\} = \left(X^T R^{-1} X + Q^{-1}\right)^{-1}$$

From the above equation, we see that the best linear unbiased (Gauss-Markov) estimator is a limiting case of the minimum variance estimator with $Q^{-1} = 0$.

3.2.4. Recursive Least Squares

The least squares techniques discussed above assume that all of the data (X and y) is available as one batch of data of length N . If we have a model where the output is a function of past inputs and outputs (e.g., an ARX model) we can use **recursive least squares** to reformulate the problem so that a parameter estimate can be made with sequential data stream. By separately considering the terms $X^T W X$ and $X^T W y$, we can explicitly write these matrix multiplications as the sums that they are. Separating out the most recent data point and applying the matrix inversion lemma to prevent the continual inversion of a matrix,² we can provide the following "update" formulas. In the recursive process for weighted least squares, some assumptions must be made to the weight applied to the last measurement. We consider ordinary least squares and two weighting assumptions: weighted least squares with the weight defined as $w = a\gamma^{n-k}$ and weighted least squares with $w(k, m) = \lambda(k)w(k-1, m)$ and $1 \leq m \leq n-1$.

3.2.4.1. Ordinary and Weighted Recursive Least Squares

Since $a = \gamma = 1$ is ordinary least squares, the following algorithm is suitable for both ordinary and exponentially weighted least squares. The regressor variables are updated by initially forming $X(k+1)$ from the first n measurements - which remains an $n \times p$ matrix containing the last $p = m$ inputs + n outputs - selecting an initial condition for $\hat{\theta}_{WLS}$ and $P(k)$ then using:

$$L(k+1) = \frac{P(k)}{\gamma} X(k+1) \left(a^{-1} + X^T(k+1) \frac{P(k)}{\gamma} X^T(k+1) \right)^{-1}$$

with

$$\hat{\theta}_{WLS}(k+1) = \hat{\theta}_{WLS}(k) + L(k+1) \left(y(k+1) - X^T(k+1) \hat{\theta}_{WLS}(k) \right)$$

to update the predictor, where P is updated with

$$P(k+1) = \frac{1}{\gamma} \left(I - L(k+1) X^T(k+1) \right) P(k)$$

Initial conditions can be determined from an initial batch estimate or an estimate of $\hat{\theta}_{WLS}$ with $P(0)$ "large."

² $(A + BCD)^{-1} = A^{-1} - A^{-1}B(C^{-1} + DA^{-1}B)^{-1}DA^{-1}$

3.2.4.2. Weighted Recursive Least Squares

While the parameter update equation remains the same, the assumption that $w(k, m) = \lambda(k)w(k-1, m)$ and $1 \leq m \leq n-1$ changes the update equations for L and P as follows:

$$L(k+1) = P(k)X(k+1)(\lambda(k+1)W^{-1} + X^T(k+1)P(k)X^T(k+1))^{-1}$$
$$P(k+1) = \frac{P(k) - P(k)X(k+1)(\lambda(k+1)W^{-1} + X^T(k+1)P(k)X^T(k+1))^{-1}X^T(k+1)P(k)}{\lambda(k+1)}$$

3.2.5. Ridge Regression

The aim of another modification of ordinary least squares - ridge regression - is the reduction of the mean square error [12]. This is accomplished by the addition of a symmetric matrix K:

$$\hat{\theta}_R = (X^T R^{-1} X + K)^{-1} X^T R^{-1} Y$$

The reduction in the m.s.e. comes about because the addition of the matrix K reduces any ill conditioning in the regressor matrix by preventing any of the singular values of the regressor from being very small.

3.2.6. Chi-Square Fitting

In this case, we assume that each data point y_i has a measurement error that is independently random and distributed as a normal distribution around the true model. Suppose that the standard deviation is the same for all points; then the probability of the data set is the product of probabilities of each point [13]:

$$P = \prod \left\{ \exp \left[-\frac{1}{2} \left(\frac{y_i - y(x_i)}{\sigma} \right)^2 \right] \Delta y \right\}$$

Maximizing this is equivalent to maximizing its logarithm, or minimizing the negative of its logarithm.

$$\left[\sum \frac{[y_i - y(x_i)]^2}{2\sigma^2} \right] - N \log \Delta y$$

Since N, σ , and Δy are all constants, minimizing this equation is equivalent to minimizing:

$$\sum_{i=1}^N [y_i - y(x_i; \theta_1 \dots \theta_M)]^2$$

and that least squares fitting is maximum likelihood estimation of the fitted parameters.

If each data point has its own standard deviation, however, the probability of the data set is modified by considering σ_i in place of σ . This changes the maximum likelihood estimate to:

$$\chi^2 \equiv \sum_{i=1}^N \left[\frac{y_i - y(x_i; \theta_1 \dots \theta_M)}{\sigma_i} \right]^2$$

Once we have adjusted $a_1 \dots a_m$ to minimize that χ^2 , the terms in the sum are no longer statistically independent. However, the probability distribution for different values of χ^2 at its minimum is the chi-square distribution for $N-M$ degrees of freedom. Therefore:

$$\begin{aligned} Q(\chi^2 | v) &= Q\left(\frac{v}{2} \mid \frac{\chi^2}{2}\right) = Q\left(\frac{v}{2}, \frac{\chi^2}{2}\right) \\ &= 1 - P\left(\frac{v}{2}, \frac{\chi^2}{2}\right) = \frac{\Gamma\left(\frac{v}{2}, \frac{\chi^2}{2}\right)}{\Gamma(a)} \\ &= \frac{1}{\Gamma(a)} \int_x^\infty e^{-t} t^{a-1} dt \end{aligned}$$

gives the probability Q that the chi-square should exceed a particular value χ^2 by chance, where $v = N - M$ is the number of degrees of freedom. If Q is a very small probability, then the apparent discrepancies are unlikely to be chance fluctuations. Then either (1) the model is wrong, (2) the measurement errors are larger than stated, or (3) the measurement errors are not normally distributed.

As a rule of thumb, a typical value of χ^2 for a moderately good fit is $\chi^2 \approx v$ since for asymptotically large v , the statistic χ^2 becomes normally distributed with mean v and standard deviation $\sqrt{2v}$.

Given a measurement error, the design matrix can be defined as:

$$A_{ij} = \frac{X_j(x_i)}{\sigma_i}$$

basis functions
 $X_1() \quad X_2() \quad \dots X_M()$

$$A_{ij} = \frac{X_j(x_i)}{\sigma_i} \Rightarrow \begin{array}{ccccc} & x_1 & \frac{X_1(x_1)}{\sigma_1} & \frac{X_2(x_1)}{\sigma_1} & \dots \\ & x_2 & \frac{X_1(x_2)}{\sigma_2} & \frac{X_2(x_2)}{\sigma_2} & \dots \end{array}$$

where σ_i is the measurement error and the output matrix is $c_i = \frac{y_i}{\sigma_i}$.

The solution by SVD where ω_i are the singular values, yields $\theta = \sum_{i=1}^M \frac{U_{(i)} \cdot c}{\omega_i} V_{(i)}$ and $\text{Cov}(\theta_j, \theta_k) = \sum_{i=1}^M \frac{V_{ji} V_{ki}}{\omega_i^2}$.

3.3. Eigenstructure Realization Algorithm (ERA)

Introduced in 1985, this technique has enjoyed wide application in the identification of modal parameters for lightly damped space structures [14,15,16,17]. The algorithm we present comes from [18] and includes an observer that provides the capability to identify a state space realization of a stochastic system.

Consider a discrete, time-invariant multivariable linear system:

$$\begin{aligned} x_{t_{i+1}} &= A x(t_i) + B u(t_i) + M w_d(t_i) \\ y(t_i) &= C x(t_i) + D u(t_i) + v(t_i) \end{aligned}$$

If we combine the two equations, assume a noise free (deterministic) system and zero initial conditions, we can write the system as: $y = Y U$ ($x(i) \in \mathbb{R}^n, y(i) \in \mathbb{R}^m, u(i) \in \mathbb{R}^r$).

$$y = [y(0) \ y(1) \ y(2) \ \cdots \ y(k-1)]$$

$$U = \begin{bmatrix} u(0) & u(1) & u(2) & \cdots & u(k-1) \\ & u(0) & u(1) & \cdots & u(k-2) \\ & & u(0) & \cdots & u(k-3) \\ & & & \ddots & \vdots \\ & & & & u(0) \end{bmatrix}$$

and

$$Y = [D \ CB \ CAB \ \cdots \ CA^{k-2}B]$$

The elements of the matrix Y are called the Markov parameters: $D \ CB \ CAB \ \cdots \ C^{k-2}B$.

Looking at the dimensions, we have $l \times rN$ unknowns (m outputs, r inputs, and N data samples) and only $l \times N$ equations. For a finite dimensional linear system, Y is unique; however, for the case where $r > 1$, the solution to the above system is not unique. Therefore, without modification, this system cannot be used to identify unknown Markov parameters. Assume, however, that the system is asymptotically stable. Then, for a sufficiently large p , $A^k \approx 0$ for $i \geq p$ and the system can be approximated by a truncated version of U and Y .

Actually, an observer for the above system can be developed that will be as stable as desired (refer to the linear discrete-time predictor from Chapter 5) and the resulting

Markov parameters will be the Markov parameters of the observer. The system Markov parameters will have to be extracted from the observer parameters.

The major assumption here is that of ergodicity. This is the assumption that the time average of a stationary random process is identical to an ensemble average of the same process and allows the interchange of the time average and the expected value³.

The form of equations remains the same, with the following matrices modified to include the effects of the Kalman gain and nonzero initial conditions. Choose p such that $mp > n$ (where n is the number of states) and beginning at the $p+1$ measurement:

$$y = [y(p+1) y(p+2) y(p+3) \cdots y(k-1)]$$

$$\bar{Y} = [D \quad C\bar{B} \quad C\bar{A}\bar{B} \cdots C\bar{A}^{k-1}\bar{B}]$$

with

$$\bar{A} = A + MC$$

$$\bar{B} = [B + MD, -M]$$

and

$$U = \begin{bmatrix} \begin{bmatrix} u(p+1) \\ u(p) \\ y(p) \end{bmatrix} & \begin{bmatrix} u(p+2) \\ u(p+1) \\ y(p+1) \end{bmatrix} & \begin{bmatrix} u(p+3) \\ u(p+2) \\ y(p+2) \end{bmatrix} & \cdots & \begin{bmatrix} u(k-1) \\ u(k-2) \\ y(k-2) \end{bmatrix} \\ \begin{bmatrix} u(p-1) \\ y(p-1) \end{bmatrix} & \begin{bmatrix} u(p) \\ y(p) \end{bmatrix} & \begin{bmatrix} u(p+1) \\ y(p+1) \end{bmatrix} & \cdots & \begin{bmatrix} u(k-3) \\ y(k-3) \end{bmatrix} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \begin{bmatrix} u(0) \\ y(0) \end{bmatrix} & \begin{bmatrix} u(1) \\ y(1) \end{bmatrix} & \begin{bmatrix} u(2) \\ y(2) \end{bmatrix} & \cdots & \begin{bmatrix} u(k-p-2) \\ y(k-p-2) \end{bmatrix} \end{bmatrix}$$

When $C\bar{A}^k\bar{B} \approx 0$ for $i \geq p$, the system $y = \bar{Y} U$ can be solved for \bar{Y} with real data using a **weighted least squares**. Once the observer Markov parameters are determined, the system parameters must be extracted. First, partition $\bar{Y} = [\bar{Y}_{-1} \quad \bar{Y}_0 \quad \bar{Y}_1 \cdots \bar{Y}_{p-1}]$. Now, the general relationship between the observer and the system is:

$$Y_k = \bar{Y}_k^{(1)} + \sum_{i=0}^{k-1} \bar{Y}_i^{(2)} Y_{k-i-1} + \bar{Y}^{(2)} D$$

with $\bar{Y}_{-1} = D$ and

$$\bar{Y}_k^{(1)} = C(A + KC)^k (B + KD)$$

$$\bar{Y}_k^{(2)} = -C(A + KC)^k K$$

There are only $p+1$ observer Markov parameters with $\bar{Y}_k^{(1)}$ and $\bar{Y}_k^{(2)} = 0$ for $k > p$.

³An ensemble average is an average of different realizations of the same stochastic process.

Now that we have extracted the system Markov parameters from the observer, we can recover the state space model by the ERA. Define the following $r_1 \times s$ block data matrix:

$$H(\tau) = \begin{bmatrix} Y_\tau & Y_{\tau+1} & Y_{\tau+2} & \cdots & Y_{\tau+s-1} \\ Y_{\tau+1} & Y_{\tau+2} & Y_{\tau+3} & \cdots & Y_{\tau+s} \\ Y_{\tau+2} & Y_{\tau+3} & Y_{\tau+4} & \cdots & Y_{\tau+s+1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ Y_{\tau+r_1-1} & Y_{\tau+r_1} & Y_{\tau+r_1+1} & \cdots & Y_{\tau+r_1+s-2} \end{bmatrix}$$

The order of the system is determined by the singular value decomposition of $H(0)$,

$$H(0) = U \Sigma V^T = U_1 S_1 V_1^T$$

where Σ are all of the singular values. S_1 is an $n \times n$ diagonal matrix of positive singular values that are retained and n will become the order of the system:

$$A = S_1^{-1/2} U_1^T H(1) V_1 S_1^{-1/2}$$

$$B = S_1^{-1/2} V_1 E_m$$

$$C = E_r^T U_1 S_1^{-1/2}$$

where $E_r^T = \begin{bmatrix} I_{r \times r} & 0_{r \times (r_1-m)m} \end{bmatrix}$ and $E_m^T = \begin{bmatrix} I_{m \times m} & 0_{m \times (r_1-m)m} \end{bmatrix}$.

The observer gain can be extracted in a similar fashion. First we must recover the sequence of parameters $Y_k^0 = CA^k K$ with the general relationship:

$$Y_k^0 = -\bar{Y}_k^{(2)} + \sum_{i=0}^{k-1} \bar{Y}_i^{(2)} Y_{k-i-1}^0$$

Then the Kalman filter gain in the observer can be computed from $K = -A^{-1}M$ where M is the result of the least squares fit $M = (O^T O)^{-1} O^T Y^0$ with $Y^0 = [Y_0^0 \ Y_1^0 \ Y_2^0 \ \cdots \ Y_k^0]^T$ and $O = [C \ CA \ CA^2 \ \cdots \ CA^k]^T$.

To summarize:

1. Choose p at least 4 or 5 times larger than the order of the system.
2. Form y and U and compute the least squares solution for the parameter vector \bar{Y} .
3. Recover the combined system and Kalman filter Markov parameters. To solve for more system parameters than the number of observer parameters, simply set the extra observer parameters to zero.
4. Realize the state space model using ERA.
5. Determine the Kalman gain from the system parameters and C A and Y^0 .

3.3. Autoregressive Time Series Models

Consider a linear time-invariant complete dynamical system. The universe $U = (R^q)^Z$, the model class is \mathcal{L}^q . For any data set $D \subseteq (R^q)^Z$ -- any family of q -vector time-series -- the MPUM exists. The MPUM may be parametrized as a polynomial matrix $R(s, s^{-1})$ in an AR representation, by $(P(s, s^{-1}), Q(s, s^{-1}), \Pi)$ in an I/O representation, by matrices (E, F, G) in a state representation, or by (A, B, C, D, Π) in an I/S/O representation [19].

Given data $Z \rightarrow R^q$, form the Hankel matrix of the data

$$h = \begin{bmatrix} \cdots & D(-1) & D(0) & D(1) & \cdots & D(t') & \cdots \\ \cdots & D(0) & D(1) & D(2) & \cdots & D(t'+1) & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \cdots & D(t-1) & D(t) & D(t+1) & \cdots & D(t'+1) & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

where $D(t) = [d_1(t) \ d_2(t) \ \cdots \ d_N(t)] \in R^{q \times N}$.

Let h_t ($t \in Z_+$) denote the truncation of h to its first $q(t+1)$ rows. Consider the ranks of the h_t 's. Because of the Hankel structure of h it follows that the $\text{rank}(h_0), \text{rank}(h_1), \text{rank}(h_2), \dots, \text{rank}(h_{t+1}), \text{rank}(h_t), \dots$ is a nonincreasing sequence of nonnegative integers. Hence its limit exists and is reached in a finite number of steps. Determine $t' \in Z_+$ such that $\text{rank}(h_t) - \text{rank}(h_{t-1})$ is constant for $t' \geq t$ (define $\text{rank } h_{-1} = 0$).

Consider $h_{t'}$ and compute vectors $r_1, r_2, \dots, r_g \in R^{1 \times (t'+1)q}$ such that they span the orthogonal complement of the columns of $h_{t'}$.

Write the vectors r_j as $[r_{i,0}, r_{i,1}, \dots, r_{i,t'}]$ with each $r_{i,j} \in R^{1 \times q}$.

Form the polynomials $\tilde{r}(s) = \sum_{j=0}^{t'} r_{i,j} s^j \in R^{1 \times q}[s]$, and define

$$R(s) = \begin{bmatrix} \tilde{r}_1(s) \\ \tilde{r}_2(s) \\ \vdots \\ \tilde{r}_g(s) \end{bmatrix} \in R^{g \times q}[s]$$

then $R(\sigma) = 0$ is an AR model for the MPUM.

4. CORRELATION APPROACHES

Ideally the prediction error $\varepsilon(N, \hat{\theta})$ for a "good" model should be independent of past data Z^{N-1} . If $\varepsilon(N, \hat{\theta})$ is correlated with past data, then there is more information available in it. A true test of the correlation of $\varepsilon(N, \hat{\theta})$ and Z^{N-1} requires testing every nonlinear transformation of $\varepsilon(N, \hat{\theta})$ with all possible function of Z^{N-1} . This is not feasible.

We can, however, select a finite dimensional vector sequence $\{\zeta(t)\}$ derived from Z^{N-1} and force a certain transformation of $\varepsilon(N, \hat{\theta})$ to be uncorrelated with this sequence. In general we can accomplish this by filtering the prediction errors:

$$\varepsilon_F(N, \hat{\theta}) = L(q)\varepsilon(N, \hat{\theta})$$

choosing a sequence of correlation vectors:

$$\zeta(t, \hat{\theta}) = \zeta(t, Z^{N-1}, \hat{\theta})$$

and a function: $\alpha(\varepsilon_F(N, \hat{\theta}))$ for computing:

$$f_N(\hat{\theta}, Z^N) = \frac{1}{N} \sum_{t=1}^N \zeta(t, \hat{\theta}) \alpha(\varepsilon_F(t, \hat{\theta}))$$

and then finding $\hat{\theta}_N$ such that $f_N(\hat{\theta}_N, Z^N) = 0$.

4.1. Instrumental-Variable (IV) Method

If we define $\varepsilon(N, \hat{\theta})$ above to be $\varepsilon(N, \hat{\theta}) = [y(t) - \varphi^T(t)\hat{\theta}]$, and expand the sequence of the correlation vectors to include model dependent parameters by:

$$\zeta(t, \hat{\theta}) = \zeta(t, Z^{N-1}, \hat{\theta}) = K_u(q, \hat{\theta})u(t)$$

where $K_u(q, \hat{\theta})$ is a $d \times m$ matrix filter and $L(q)$ is of dimension $p \times p$. With the $\dim \zeta(t) = \dim \hat{\theta} = d \times p$, we have the instrumental-variable (IV) method:

$$\hat{\theta}_{IV} = [\zeta(t, \hat{\theta})^T X]^{-1} \zeta(t, \hat{\theta})^T y$$

If we allow $\dim \zeta(t) > d$ and a minimum norm solution for $f_N(\hat{\theta}, Z^N)$, we have the extended IV method.

4.2. Recursive Instrumental-Variables (IV)

We begin with a weighting factor of $\beta(t,k) = \lambda^{t-k}$ which exponentially discounts old measurements. The recursive IV method to be described here has a distinctive feature of separating the system and noise parameter estimation [20]. Given $z(t_i) = \{x(t_i), y(t_i)\}$ and $\theta(t_{i-1})$, we update the instrumental variables and filter parameter:

$$\zeta(t_i, \hat{\theta}) = K_u(q, \hat{\theta})u(t_i)$$

$$L(t_i) = \frac{P(t_{i-1})\zeta(t_i, \theta)}{\lambda(t) + X^T(t)P(t_{i-1})\zeta(t_i, \theta)}$$

Calculate the innovations:

$$\tilde{z}(t_i) = [y(t) - X(t)^T \hat{\theta}(t_{i-1})]$$

Update the parameter estimate:

$$\hat{\theta}_{IV}(t_i) = \hat{\theta}(t_{i-1}) + L(t)\tilde{z}(t_i)$$

And then update the (inverse of) weighted covariance

$$P(t) = \frac{1}{\lambda(t)} \left[P(t_{i-1}) - \frac{P(t_{i-1})\zeta(t)X^T(t)P(t_{i-1})}{\lambda(t) + X^T(t)P(t_{i-1})\zeta(t)} \right]$$

5. MAXIMUM LIKELIHOOD APPROACHES

The following discussions are limited to linear-time invariant (discrete time) systems. Nonlinear effects can be included by appropriately modifying the prediction equations in either of two ways. First, nonlinear system effects can be directly included in the propagation of the state. Second, nonlinear measurements (with linear propagation) can be handled with an extended Kalman filter model. A state space model structure is assumed with independent (uncorrelated) white noise process disturbances and Gaussian measurement noise.

Maximum likelihood parameter estimators can be divided into 3 classes depending on whether measurement and process noise are accounted for [21]. All of the maximum likelihood parameter estimators that we consider account for both.

Given a linear time-invariant discrete state space model:

$$x_{t_{i+1}} = A(\theta) x(t_i) + B(\theta) u(t_i) + M(\theta) w_d(t_i)$$

with (coincident) sampled data measurements: $y_{t_i} = C(\theta) x(t_i) + D(\theta) u(t_i) + v_d(t_i)$. Note that the output model has been extended to allow for direct transmission of the input through to the output via the matrix $D(\theta)$.

The statistics of $w_d(t)$ are assumed to be $E\left\{\begin{bmatrix} w_i \\ w_i \end{bmatrix} \begin{bmatrix} w_j^T & w_j^T \end{bmatrix}\right\} = I\delta(i-j)$ with the strength of the disturbance included in M . Likewise the strength of the measurement noise is $E\left\{\begin{bmatrix} v_i \\ v_i \end{bmatrix} \begin{bmatrix} v_j^T & v_j^T \end{bmatrix}\right\} = R\delta(i-j)$.

There are a number of conditional probability density functions that could be used for the likelihood function. Variations include fixed length versus growing length functions, specification of *a priori* statistics, use of the initial conditions, and the sensitivity of the estimate on the identified parameters. The density function most appropriate is:

$$\begin{aligned} f_{x(t_i)|Z(t_i),\theta} &= f_{x(t_i)|Z(t_i),\theta} f_{Z(t_i),\theta} \\ &= f_{x(t_i)|Z(t_i),\theta} \prod_{j=1}^i f_{Z(t_j)|Z(t_{j-1}),\theta} \end{aligned}$$

Minimization of the likelihood function with this density results in the state predicted by the Kalman-Bucy filter, but there is no closed form solution to compute the partial derivatives.

5.1. Full Scale Estimator

5.1.1. Theory

We are going to develop a parameter estimator that will use the last N observations to identify v uncertain parameters in the system and input matrices A and B . (Note: Uncertainty in these parameters could not be separated from uncertainties in C and D . Consequently, the assumption is that C and D are known and the uncertainty is A and B .)

The iterative estimator for minimization of the likelihood equation:

$$\left. \frac{\partial L[\hat{\theta}, Z^N]}{\partial \hat{\theta}} \right|_{\hat{\theta}(t_i) = \theta_s(t_i)}$$

using the method of "steepest descent" is:

$$\hat{\theta}(t_i) = \hat{\theta}(t_i) - \left[\frac{\partial^2 L[\hat{\theta}, Z^N]}{\partial \hat{\theta}^2} \right]^{-1} \left[\frac{\partial L[\hat{\theta}, Z^N]}{\partial \hat{\theta}} \right]$$

To use this algorithm, the Hessian (second derivative matrix) must be of full rank. Using a technique called "scoring," we can approximate the Hessian with the conditional information matrix [22]:

$$J[t_i, \hat{\theta}(t_i)] = E \left\{ \frac{\partial L[\theta, Z^N]}{\partial \hat{\theta}} \frac{\partial L[\theta, Z^N]}{\partial \hat{\theta}} \right| \theta = \hat{\theta}(t_i) \right\}$$

which results in:

$$\hat{\theta}(t_i) = \hat{\theta}(t_i) + [J[t_i, \hat{\theta}(t_i)]]^{-1} \left[\frac{\partial L[\hat{\theta}, Z^N]}{\partial \hat{\theta}} \right]^T$$

with the score matrix defined as $[\partial L[\theta, Z^N] / \partial \hat{\theta}]$ and computed from:

$$\frac{\partial L}{\partial \hat{\theta}_k} [\hat{x}(t_i), \hat{\theta}(t_i), Z_i] = \gamma_k [Z_j, \hat{\theta}(t_i)] + \sum_{j=i-N+1}^i S_k^j [Z_i, \hat{\theta}(t_i)]$$

with

$$\begin{aligned} S_k^j [Z_j, \hat{\theta}(t_i)] &= \frac{\partial \hat{x}(t_j^-)^T}{\partial \hat{\theta}_k} C(t_j)^T O(t_j)^{-1} r_j \\ &\quad - \frac{1}{2} \text{tr} \left\{ [O(t_j)^{-1} - O(t_j)^{-1} \bar{z}_j \bar{z}_j^T O(t_j)^{-1}] \frac{\partial O(t_j^-)^T}{\partial \hat{\theta}_k} \right\} \\ \gamma_k [Z_i, \hat{\theta}(t_i)] &= -\frac{1}{2} \text{tr} \left\{ P(t_i^+) \frac{\partial P(t_i H^+)}{\partial \hat{\theta}_k} \right\} \end{aligned}$$

The conditional information matrix can also be decomposed into a sum of the N most recent measurements:

$$J_k[t_i, \hat{x}(t_i), \hat{\theta}(t_i)] = E\left\{\gamma_k[Z_i, \hat{\theta}(t_i)]\gamma_k^T[Z_i, \hat{\theta}(t_i)]\right\} \\ + \sum_{j=i-N+1}^i E\left\{s_k^1[Z_j, \hat{\theta}(t_j)]s_k^1[Z_j, \hat{\theta}(t_j)]^T\right\}$$

where

$$E\left\{s_k^1[Z_j, \hat{\theta}(t_j)]s_k^1[Z_j, \hat{\theta}(t_j)]^T\right\} = \frac{1}{2}\text{tr}\left[O(t_j)^{-1}\frac{\partial O(t_j)}{\partial \hat{\theta}_k}O(t_j)^{-1}\frac{\partial O(t_j)}{\partial \hat{\theta}_1} \right. \\ \left. + 2O(t_j)^{-1}C(t_j)E\left\{\frac{\partial \hat{x}(t_j)}{\partial \hat{\theta}_k}\frac{\partial \hat{x}(t_j)^T}{\partial \hat{\theta}_1}\right\}C(t_j)^T\right] \\ E\left\{\gamma_k[Z_i, \hat{\theta}(t_i)]\gamma_k^T[Z_i, \hat{\theta}(t_i)]\right\} = \frac{1}{2}\text{tr}\left[P(t_i^+)^{-1}\frac{\partial P(t_i^+)}{\partial \hat{\theta}_k}P(t_i^+)^{-1}\frac{\partial P(t_i^+)}{\partial \hat{\theta}_1} \right. \\ \left. + 2P(t_i^+)^{-1}E\left\{\frac{\partial \hat{x}(t_i^+)}{\partial \hat{\theta}_k}\frac{\partial \hat{x}(t_i^+)^T}{\partial \hat{\theta}_1}\right\}\right]$$

5.1.2. Algorithm

Considering the propagation of the values in time, incorporation of measurements, and the summation over the last N residuals, the implementation of the above equations is quite complex. To make the programming tractable (primarily to explicitly define the time steps), the complete algorithm (including previous equations on the Kalman filter) will be presented here.

Beginning at time (t_i) we have $\hat{x}(t_{i-1})$ and $P(t_{i-1})$. However the above sums consider the last N measurements, so that only $\hat{x}(t_{i-N})$ and $P(t_{i-N})$ are fixed as initial conditions. Using $j = (i - N + 1), (i - N + 2), \dots, (i)$, we will recompute $\hat{x}(t_j)$ and $P(t_j)$ from (t_{i-N+1}) to (t_i) .

For each of the N measurements:

1. First the states and equations for the score and conditional information matrix are propagated in time:

A. State estimate extrapolation:

$$\hat{x}(t_j^-) = A(t_{j-1})\hat{x}(t_{j-1}^+) + \int_{t_{j-1}}^{t_j} \Phi(t_j, \tau)B(\tau)u(\tau)d\tau$$

B. Error covariance extrapolation:

$$P(t_j^-) = A(t_{j-1})P(t_{j-1}^+)A^T(t_{j-1}) + Q_d(t_{j-1})$$

C. Filter gain calculation:

$$K(t_j) = P(t_j)C^T(t_j)[O(t_j)]^{-1}$$

with:

$$O(t_j) = C(t_j)P(t_j^-)C^T(t_j) + R_d(t_j)$$

D. Propagate the v (uncertain parameter) "score equations":

$$\frac{\partial \hat{x}(t_j^-)}{\partial \hat{\theta}_k} = A(t_j, t_{j-1}) \frac{\partial \hat{x}(t_{j-1}^+)}{\partial \hat{\theta}_k} + \frac{\partial A(t_j, t_{j-1})}{\partial \hat{\theta}_k} x(t_{j-1}^+) + \frac{\partial B(t_{j-1})}{\partial \hat{\theta}_k} u(t_{j-1})$$

$$\frac{\partial P(t_j^-)}{\partial \hat{\theta}_k} = \frac{\partial A(t_j, t_{j-1})}{\partial \hat{\theta}_k} P(t_{j-1}^+) A^T(t_j, t_{j-1}) + A(t_j, t_{j-1}) P(t_{j-1}^+) \frac{\partial A^T(t_j, t_{j-1})}{\partial \hat{\theta}_k} + A(t_j, t_{j-1}) \frac{\partial P(t_{j-1}^+)}{\partial \hat{\theta}_k} A^T(t_j, t_{j-1})$$

$$\frac{\partial O(t_i)}{\partial \hat{\theta}_k} = O(t_i) \frac{\partial P(t_i^-)}{\partial \hat{\theta}_k} O^T(t_i)$$

E. While terms for the conditional information matrix are propagated by:

$$\begin{aligned} & E \left\{ \frac{\partial \hat{x}(t_j^-)}{\partial \hat{\theta}_k} \frac{\partial \hat{x}(t_j^-)}{\partial \hat{\theta}_l}^T \right\} \\ &= A(t_{j-1}) E \left\{ \frac{\partial \hat{x}(t_{j-1}^+)}{\partial \hat{\theta}_k} \frac{\partial \hat{x}(t_{j-1}^+)}{\partial \hat{\theta}_l}^T \right\} A^T(t_{j-1}) + \frac{\partial A(t_{j-1})}{\partial \hat{\theta}_k} E \left\{ \hat{x}(t_{j-1}^+) \hat{x}(t_{j-1}^+)^T \right\} \frac{\partial A^T(t_{j-1})}{\partial \hat{\theta}_l} \\ &+ A(t_{j-1}) E \left\{ \frac{\partial \hat{x}(t_{j-1}^+)}{\partial \hat{\theta}_k} \hat{x}(t_{j-1}^+)^T \right\} \frac{\partial A^T(t_{j-1})}{\partial \hat{\theta}_l} + \frac{\partial A(t_{j-1})}{\partial \hat{\theta}_k} E \left\{ \hat{x}(t_{j-1}^+) \frac{\partial \hat{x}^T(t_{j-1}^+)}{\partial \hat{\theta}_l} \right\} A^T(t_{j-1}) \\ &+ \frac{\partial B_d(t_{j-1})}{\partial \hat{\theta}_k} E \left\{ u(t_{j-1}) u(t_{j-1})^T \right\} \frac{\partial B_d(t_{j-1})}{\partial \hat{\theta}_l} \\ &+ \frac{\partial A(t_{j-1})}{\partial \hat{\theta}_k} E \left\{ \hat{x}(t_{j-1}^+) u(t_{j-1})^T \right\} \frac{\partial B_d(t_{j-1})}{\partial \hat{\theta}_l} + \frac{\partial B_d(t_{j-1})}{\partial \hat{\theta}_k} E \left\{ u(t_{j-1}) \hat{x}(t_{j-1}^+)^T \right\} \frac{\partial B_d(t_{j-1})}{\partial \hat{\theta}_l} \\ &+ A(t_{j-1}) E \left\{ \frac{\partial \hat{x}(t_{j-1}^+)}{\partial \hat{\theta}_k} u(t_{j-1})^T \right\} \frac{\partial B_d(t_{j-1})}{\partial \hat{\theta}_l} + \frac{\partial B_d(t_{j-1})}{\partial \hat{\theta}_k} E \left\{ u(t_{j-1}) \frac{\partial \hat{x}^T(t_{j-1}^+)}{\partial \hat{\theta}_l} \right\} A^T(t_{j-1}) \end{aligned}$$

and

$$\begin{aligned}
E\{\hat{\mathbf{x}}(t_j^-)\hat{\mathbf{x}}^T(t_j^-)\} &= \mathbf{A}(t_{j-1})E\{\hat{\mathbf{x}}(t_{j-1}^+)\hat{\mathbf{x}}^T(t_{j-1}^+)\}\mathbf{A}^T(t_{j-1}) \\
&\quad + \mathbf{B}_d(t_{j-1})E\{\mathbf{u}(t_{j-1})\mathbf{u}(t_{j-1})^T\}\mathbf{B}_d^T(t_{j-1}) \\
&\quad + \mathbf{A}(t_{j-1})E\{\hat{\mathbf{x}}(t_{j-1}^+)\mathbf{u}(t_{j-1})^T\}\mathbf{B}_d^T(t_{j-1}) \\
&\quad + \mathbf{B}_d(t_{j-1})E\{\mathbf{u}(t_{j-1})\hat{\mathbf{x}}^T(t_{j-1}^+)\}\mathbf{A}^T(t_{j-1})
\end{aligned}$$

and

$$\begin{aligned}
&E\left\{\frac{\partial \hat{\mathbf{x}}(t_j^-)}{\partial \hat{\boldsymbol{\theta}}_k} \hat{\mathbf{x}}(t_j^-)^T\right\} \\
&= \mathbf{A}(t_{j-1})E\left\{\frac{\partial \hat{\mathbf{x}}(t_{j-1}^+)}{\partial \hat{\boldsymbol{\theta}}_k} \hat{\mathbf{x}}(t_{j-1}^+)^T\right\}\mathbf{A}^T(t_{j-1}) + \frac{\partial \mathbf{A}(t_{j-1})}{\partial \hat{\boldsymbol{\theta}}_k} E\{\hat{\mathbf{x}}(t_{j-1}^+)\hat{\mathbf{x}}(t_{j-1}^+)^T\}\mathbf{A}^T(t_{j-1}) \\
&\quad + \mathbf{A}(t_{j-1})E\left\{\frac{\partial \hat{\mathbf{x}}(t_{j-1}^+)}{\partial \hat{\boldsymbol{\theta}}_k} \mathbf{u}(t_{j-1})^T\right\}\mathbf{B}_d(t_{j-1}) + \frac{\partial \mathbf{B}_d(t_{j-1})}{\partial \hat{\boldsymbol{\theta}}_k} E\{\mathbf{u}(t_{j-1})\mathbf{u}(t_{j-1})^T\}\mathbf{B}_d^T(t_{j-1}) \\
&\quad + \frac{\partial \mathbf{A}(t_{j-1})}{\partial \hat{\boldsymbol{\theta}}_k} E\{\hat{\mathbf{x}}(t_{j-1}^+)\mathbf{u}(t_{j-1})^T\}\mathbf{B}_d(t_{j-1}) + \frac{\partial \mathbf{B}_d(t_{j-1})}{\partial \hat{\boldsymbol{\theta}}_k} E\{\mathbf{u}(t_{j-1})\hat{\mathbf{x}}(t_{j-1}^+)^T\}\mathbf{A}^T(t_{j-1})
\end{aligned}$$

and

$$\begin{aligned}
E\{s_k^l[Z_j, \hat{\boldsymbol{\theta}}(t_i)]s_i^l[Z_j, \hat{\boldsymbol{\theta}}(t_i)]\} &= \frac{1}{2}\text{tr}\left[O(t_j)^{-1}\frac{\partial O(t_j)}{\partial \hat{\boldsymbol{\theta}}_k}O(t_j)^{-1}\frac{\partial O(t_j)}{\partial \hat{\boldsymbol{\theta}}_l}\right. \\
&\quad \left.+ 2O(t_j)^{-1}C(t_j)E\left\{\frac{\partial \hat{\mathbf{x}}(t_j^-)}{\partial \hat{\boldsymbol{\theta}}_k}\frac{\partial \hat{\mathbf{x}}(t_j^-)^T}{\partial \hat{\boldsymbol{\theta}}_l}\right\}C(t_j)\right]
\end{aligned}$$

Maintain a running sum for each k and l :

$$\sum_{j=i-N+1}^i E\{s_k^l[Z_j, \hat{\boldsymbol{\theta}}(t_j)]s_i^l[Z_j, \hat{\boldsymbol{\theta}}(t_j)]\}$$

2. Having propagated the terms forward in time, the measurement is incorporated by:

A. Compute the residual:

$$\tilde{\mathbf{z}}(t_j^+) = [\mathbf{z}(t_j) - \mathbf{H}(t_j)\hat{\mathbf{x}}(t_j^-)]$$

B. State estimate update:

$$\hat{\mathbf{x}}(t_j^+) = \hat{\mathbf{x}}(t_j^-) + \mathbf{K}(t_j)\tilde{\mathbf{z}}(t_j)$$

C. Error covariance update:

$$\mathbf{P}(t_j^+) = \mathbf{U}(t_j)\mathbf{P}(t_j^-)\mathbf{U}^T(t_j) + \mathbf{K}(t_j)\mathbf{R}(t_j)\mathbf{K}^T(t_j)$$

with

$$U(t_j) = I - K(t_j)C(t_j)$$

D. Define the following terms:

$$n_j = O^{-1}(t_j)\tilde{z}(t_j)$$

$$V(t_j) = O^{-1}(t_j) - n_j n_j^T$$

E. Update the score equations (with $k = 1, 2, \dots, v$):

$$s_k^1[Z_j, \hat{\theta}(t_j)] = \frac{\partial \hat{x}^T(t_j^-)}{\partial \hat{\theta}_k} [C^T(t_j)n_j] - \frac{1}{2} \text{tr} \left\{ V(t_j)^{-1} \frac{\partial O(t_j)}{\partial \hat{\theta}_k} \right\}$$

$$\frac{\partial \hat{x}(t_j^+)}{\partial \hat{\theta}_k} = U(t_j) \left\{ \frac{\partial \hat{x}(t_j^-)}{\partial \hat{\theta}_k} + \frac{\partial P(t_j^-)}{\partial \hat{\theta}_k} [C^T(t_j)n_j] \right\}$$

$$\frac{\partial P(t_j^+)}{\partial \hat{\theta}_k} = U(t_j) \frac{\partial P(t_j^-)}{\partial \hat{\theta}_k} U^T(t_j)$$

Maintain the running sum for each k :
$$\sum_{j=i-N+1}^i s_k^1[Z_j, \hat{\theta}(t_j)]$$

F. Update the conditional information matrix relations:

$$E \left\{ \frac{\partial \hat{x}(t_j^+)}{\partial \hat{\theta}_k} \frac{\partial \hat{x}(t_j^+)}{\partial \hat{\theta}_1}^T \right\} = U(t_j) E \left\{ \frac{\partial \hat{x}(t_j^-)}{\partial \hat{\theta}_k} \frac{\partial \hat{x}(t_j^-)}{\partial \hat{\theta}_1}^T \right\} U^T(t_j)$$

$$+ U(t_j) \frac{\partial P(t_j^-)}{\partial \hat{\theta}_k} C^T(t_j) O^{-1}(t_j) C(t_j) \frac{\partial P(t_j^-)}{\partial \hat{\theta}_1} U(t_j)$$

$$E \{ \hat{x}(t_j^+) \hat{x}^T(t_j^+) \} = E \{ \hat{x}(t_j^-) \hat{x}^T(t_j^-) \} + K(t_j) O(t_j) K^T(t_j)$$

$$E \left\{ \frac{\partial \hat{x}(t_j^+)}{\partial \hat{\theta}_k} \hat{x}(t_j^+)^T \right\} = U(t_j) E \left\{ \frac{\partial \hat{x}(t_j^-)}{\partial \hat{\theta}_k} \hat{x}(t_j^-)^T \right\} + \frac{\partial P(t_j^-)}{\partial \hat{\theta}_k} C^T(t_j) K^T(t_j)$$

At the end of the N-step recursion, each of the following will have been computed: $\hat{x}(t_i^+)$, $P(t_i^+)$, $\frac{\partial \hat{x}(t_i^+)}{\partial \theta_k}$, $\frac{\partial P(t_i^+)}{\partial \theta_k}$, and $E\left\{\left[\frac{\partial \hat{x}(t_i^+)}{\partial \theta_k}\right]\left[\frac{\partial \hat{x}(t_i^+)}{\partial \theta_l}\right]^T\right\}$ along with the running sums $\sum_{j=i-N+1}^i E\left\{s_k^l[Z_j, \hat{\theta}(t_j)]s_l^l[Z_j, \hat{\theta}(t_j)]\right\}$ and $\sum_{j=i-N+1}^i s_k^l[Z_j, \hat{\theta}(t_j)]$.

Therefore, compute the following terms:

$$\gamma_k[Z_i, \hat{\theta}(t_i)] = -\frac{1}{2} \text{tr} \left\{ P(t_i^+) \frac{\partial P(t_i, H+)}{\partial \hat{\theta}_k} \right\}$$

and

$$E\left\{\gamma_k[Z_i, \hat{\theta}(t_i)]\gamma_l[Z_i, \hat{\theta}(t_i)]\right\} = \frac{1}{2} \text{tr} \left[P(t_i^+)^{-1} \frac{\partial P(t_i^+)}{\partial \hat{\theta}_k} P(t_i^+)^{-1} \frac{\partial P(t_i^+)}{\partial \hat{\theta}_l} \right. \\ \left. + 2P(t_i^+)^{-1} E\left\{ \frac{\partial \hat{x}(t_i^+)}{\partial \hat{\theta}_k} \frac{\partial \hat{x}(t_i^+)}{\partial \hat{\theta}_l}^T \right\} \right]$$

These matrices are added to the sums from the recursion to compute the score $\left[\frac{\partial L[\theta, Z^N]}{\partial \hat{\theta}}\right]$ and conditional information matrix $J_{kl}[t_i, \hat{x}(t_i), \hat{\theta}(t_i)]$ which are used to update the parameter estimate via the following equations:

$$\frac{\partial L}{\partial \hat{\theta}_k}[\hat{x}(t_i), \hat{\theta}(t_i), Z_i] = \gamma_k[Z_i, \hat{\theta}(t_i)] + \sum_{j=i-N+1}^i s_k^l[Z_j, \hat{\theta}(t_j)]$$

$$J_{kl}[t_i, \hat{x}(t_i), \hat{\theta}(t_i)] = E\left\{\gamma_k[Z_i, \hat{\theta}(t_i)]\gamma_l[Z_i, \hat{\theta}(t_i)]\right\} \\ + \sum_{j=i-N+1}^i E\left\{s_k^l[Z_j, \hat{\theta}(t_j)]s_l^l[Z_j, \hat{\theta}(t_j)]\right\}$$

$$\hat{\theta}(t_i) = \hat{\theta}(t_i) + \left[J[t_i, \hat{\theta}(t_i)]\right]^{-1} \left[\frac{\partial L[\hat{\theta}, Z^N]}{\partial \hat{\theta}}\right]^T$$

5.2. Modified Maximum Likelihood (MMLE)

Even with scoring and the approximations, the full scale estimator requires a large number of calculations. In the modified maximum likelihood formulation, A, B, C, D, and M are estimated and used with P to determine the Kalman gain K using an approximation based on the Ricatti equation. To provide a parameter estimator, consider the measurement equation. Since we have assumed a Gaussian error model, the CPDF for the measurement becomes:

$$P(z_i | z_{1:i-1}, \theta) = \frac{1}{[(2\pi)^m \det P]^{1/2}} \exp \left\{ -\frac{1}{2} \tilde{z}_i^T (P)^{-1} \tilde{z}_i \right\}$$

where $P = E\{\tilde{z}\tilde{z}^T\}$ with dimension $m \times m$ and $\tilde{z}_i = z_i - \hat{z}$ is the innovations process (residuals) computed by the Kalman filter (Chapter 5) (where all of the matrices could be functions of θ).

Assuming the innovations covariance is constant, use of a steady state filter results in a constant filter gain and innovations covariance. This allows the CPDF to be written as:

$$P(z|\theta) = \prod_{i=1}^N \frac{1}{[(2\pi)^m \det P]^{1/2}} \exp \left\{ -\frac{1}{2} \tilde{z}_i^T (P)^{-1} \tilde{z}_i \right\}$$

5.2.1. Maximum Likelihood (ML) Estimation

Given the above CPDF, the ML LLF becomes

$$LLF(\hat{\theta}) = \frac{1}{2} \sum_{i=1}^N \left\{ \tilde{z}_i^T (P)^{-1} \tilde{z}_i \right\} + \frac{N}{2} \log \det(P) + \frac{Nm}{2} \log 2\pi$$

A necessary condition at the minimum is that $P = E\{\tilde{z}\tilde{z}^T\}$ must equal the sample innovations covariance $\hat{P} = \frac{1}{N} \sum_{i=1}^N \tilde{z}_i \tilde{z}_i^T$ [23]. Therefore, since P has dimension $m \times m$, the first term in the LLF becomes $Nm / 2$, and the minimization reduced to a minimization of the determinant of the sample innovations covariance matrix.

When P is known, the LLF can be minimized by minimizing the following cost function:

$$J(\hat{\theta}) = \frac{1}{2} \sum_{i=1}^N \left\{ \tilde{z}_i^T (P)^{-1} \tilde{z}_i \right\}$$

This minimization is usually carried out using a Gauss-Newton method using the first and second gradients of the cost function.

5.2.2. Maximum A Posteriori (MAP) Estimation

In the maximum a posteriori estimator, we continue to require that $\hat{P} = \frac{1}{N} \sum_{i=1}^N \tilde{z}_i \tilde{z}_i^T$ but recall that the LLF_{MAP} adds the term $-\log P(\hat{\theta})$.

Assuming that θ is normally distributed with a covariance Σ :

$$-\log P(\hat{\theta}) = \frac{1}{2} (\hat{\theta} - \theta)^T \Sigma^{-1} (\hat{\theta} - \theta) + \frac{1}{2} \log((2\pi)^m \det \Sigma)$$

the LLF_{MAP} becomes:

$$LLF_{MAP}(\hat{\theta}) = \frac{1}{2} \sum_{i=1}^N \tilde{z}_i^T (\hat{P})^{-1} \tilde{z}_i + \frac{1}{2} (\hat{\theta} - \theta)^T \Sigma^{-1} (\hat{\theta} - \theta)$$

which adds a quadratic term that biases the estimates toward *a priori* values.

5.2.3. Output Error Method

The output error identification method finds parameters that minimize the following cost function:

$$J(\theta) = \frac{1}{2} \sum_{i=1}^N \tilde{z}_i W \tilde{z}_i^T$$

where the estimate \hat{z}_i is the output of the model prediction equation.

With unstable plants, the measured and estimated responses will diverge if process noise, modeling error, or measurement errors are present [5].

6. OPTIMIZATION

As stated in the introduction, this research does not concentrate on the numerical methods used by the identification techniques. (The association of a numerical procedure or algorithm with an experimental setup and error criterion is one of the reasons there are so many "different" procedures and so little standardization in the field.) The equation or output error identifier forms combined with a minimum mean square error criterion, however, can reduce the inverse problem (modeling from data) to an optimization problem. Consequently, we provide a brief mention of some of these procedures.

Optimization is concerned with the minimization (or maximization) of a function. As stated above, in most cases where optimization is used for system identification, a mean square error function (criterion) is used, and we concentrate on minimizing that criterion. There are three significant limitations to optimization for system identification. First, most optimization techniques are not guaranteed to find a global (as opposed to a local) minimum. Second, these problems can be sensitive to numerical problems such as round-off error and truncation in the calculation of gradients. And third, the techniques usually require functions with continuous first and second derivatives.

Because of these problems, optimization techniques are only recommended for nonlinear problems that preclude other more structured approaches.

6.1. Problem Definition

Multi-objective optimization attempts to minimize a set of objectives simultaneously. One formulation for this problem is the goal attainment problem. This requires the construction of a set of goal values for the objective functions and uses a search method such as the sequential quadratic programming method.

Constrained optimization problems minimize a function subject to a set of constraints, **unconstrained optimization** problems minimize the objective function without constraints. While both of these setups could be used for system identification, the constrained optimization problem is more common.

The **minimax** solution minimizes the worst case values of a set of multi-variable functions. The values may or may not be subject to constraints.

Quadratic programming problems minimize quadratic cost functions of the form: $\frac{1}{2}\{x^T H x + c^T x\}$ subject to $Ax \leq b$. **Convex programming** allows the theory of local extrema for general nonlinear functionals to become global theory.

Many of the iterative techniques listed below are used as part of prediction error or maximum likelihood approaches.

6.2. Iterative Optimization Methods

The great majority of interesting optimization problems must be treated by computer methods. There are two basic approaches for resolving complex problems by numerical techniques: (1) formulate the necessary conditions for the optimal solution and solve these equations; or (2) implement a direct iterative search for the optimum [11].

Consider the equation $x = T(x)$. The method of **successive approximation** begins with an initial trial vector x_1 , computes $x_0 = T(x_0)$, and continues in this manner computing successive vectors $x_{n+1} = T(x_n)$. If T is a contraction mapping, there is a unique vector x_0 satisfying $x_0 = T(x_0)$, and furthermore x_0 can be obtained by the method of successive approximation. T is a contraction mapping if there is an $\alpha, 0 \leq \alpha < 1$ such that $\|T(x_1) - T(x_2)\| \leq \alpha \|x_1 - x_2\|$. We discuss approximations further in the following subsection.

Newton's method is an iterative technique for solving $P(x) = 0$. At each point, the graph of the function is approximated by its tangent and the approximate solution to the equation $P(x) = 0$ is taken to be the point where the tangent crosses the axis. The process defines a sequence of points according to the following recursion:

$$x_{n+1} = x_n - \frac{P(x_n)}{\frac{\partial P(x_n)}{\partial x}}$$

Newton's method amounts to successive approximation with $T(x) = x - \left[\frac{\partial P(x)}{\partial x} \right]^{-1} P(x)$.

The **Gauss-Newton** algorithm is a modification to Newton's method that approximates higher order terms with first order products [20]. The **Levenberg-Marquardt** algorithm is a compromise between the downhill gradient of steepest descent and the direction given by the Gauss-Newton method.

The above methods iterate on the equations derived as necessary conditions for an optimal solution. These methods converge only if the initial approximation is sufficiently close to the solution. Also, only local convergence is obtained. A more direct approach is to iterate in such a way as to decrease the cost functional from one step to the next. The general technique is to construct an equation of the form $x_{n+1} = x_n - \alpha_n p_n$ where p is a direction vector. The most widely used procedure for minimizing a functional (defined on a Hilbert space) is the method of **steepest descent**.

Conjugate direction methods reformulate the problem of steepest descent as a Hilbert space minimum norm problem. This set of methods include Fourier series expansions, orthogonalization of moments, and the conjugate gradient method.

Methods for explicitly solving constrained optimization problems include projection methods, the primal dual method, and penalty function methods.

The list of algorithms, and modifications thereof, continues well beyond the above discussion (modified Newton-Raphson, Davidon-Fletcher-Power, Leven-Marquardt Algorithm, etc.) but does not address the real issues of identification methods.

6.3. pEst

A minimum mean square error parameter estimator, pEst is an interactive program for the parameter estimation in nonlinear dynamic systems. This program solves a vector set of time-varying, finite-dimensional, ordinary differential equations that are separated into a continuous-time state equation and a discrete-time measurement equation:

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{f}[\mathbf{x}(t), \mathbf{u}(t), \boldsymbol{\theta}] \\ \mathbf{z}(t_i) &= \mathbf{g}[\mathbf{x}(t_i), \mathbf{u}(t_i), \boldsymbol{\theta}]\end{aligned}$$

A discrete-time feedback feature is provided in the equations of motion that is proportional to the difference between the measured and computed responses.

Using three separate minimization algorithms (steepest descent, modified Newton-Raphson, and Davidon-Fletcher-Power), pEst minimizes the following cost function:

$$J(\hat{\boldsymbol{\theta}}) = \frac{1}{2n_N n_z} \sum_{i=1}^{n_t} [\mathbf{z}(t_i) - \hat{\mathbf{z}}(t_i)]^T \mathbf{W} [\mathbf{z}(t_i) - \hat{\mathbf{z}}(t_i)]$$

where n_N = number of data points, and n_z = number of response variables.

6.4. Simulated Annealing

Appendix 3 contains a complete discussion of simulated annealing and a metal modeling method [24,25]. The remainder of this section provides the procedures to apply adaptive simulated annealing to stochastic systems.

Explicitly defined in N -dimensional space, the **generating probability density function** for adaptive simulated annealing considers each parameter individually. From Appendix 3, the random variable x^i is generated from $u^i \in U[0,1]$ by:

$$x^i = \text{sgn}(u^i - \frac{1}{2}) T_i [(1 + 1/T_i)^{|2u^i - 1|} - 1]$$

The **acceptance probability density function** uses a "Boltzmann" test. At each annealing time $k+1$, the cost functions, $C(p_{k+1}) - C(p_k)$, are compared using a uniform random generator, U in $[-1,1]$. If

$$\exp[-(C(p_{k+1}) - C(p_k)) / T_{\text{cost}}] > U$$

where T_{cost} is the “temperature” used for this test, then the new point is accepted as the new saved point for the next iteration. Otherwise, the last saved point is retained.

Because of the additional complexity, there are multiple **annealing temperature schedules**, one for the generating function and one for each parameter. Starting with temperature T_{i0} , the annealing temperature schedule (for the parameters) at the annealing time k for this generating function is:

$$T_i(k_i) = T_{i0} \exp[-c_i k_i^{1/N}]$$

The parameter c_i is controlled so that $T_{fi} = T_{i0} \exp[-m_i]$ when $k_f = \exp[n_i]$ so that

$$c_i = m_i \exp[-n_i / N]$$

where m_i and n_i are “free” parameters used to tune ASA for specific problems.

The annealing schedule for the cost temperature is developed similarly to the parameter temperatures. However, the index for reannealing the cost function, T_{cost} is determined by the number of accepted points, instead of the number of generated points as used for the parameters.

A multi-dimensional search should deal with the changing sensitivities of the different parameters. This is accomplished in ASA by periodic reannealing (rescaling the annealing time k) of the generating function to “stretch out” the range over which the relatively insensitive parameters are being searched.

The sensitivity of the parameters s_i is calculated at the current minimum value of the cost function C via $s_i = \partial C / \partial p^i$. The maximum sensitivity s_{max} is used with each parameter:

$$T'_{ik} = T_{ik} (s_{\text{max}} / s_i)$$

$$k_i \rightarrow k'_i = \left(\frac{\ln[T_{i0} / T'_{ik}]}{c_i} \right)^N$$

with T_{i0} set to unity to begin the search.

The acceptance temperature is similarly rescaled.

7. APPROXIMATION TECHNIQUES FOR IDENTIFICATION

7.1. Quasi-Linearization

This set of approximation techniques is concerned with the identification of nonlinear systems. This approach uses maximum likelihood to identify linearized dynamics. These techniques do not work well for systems that depart from linearity.

7.2. Stochastic Approximation

Stochastic approximation may be also considered as an iterative optimization (gradient) method [26]. It can be applied to any problem which can be formulated as a regression in which repeated observations are made. This approach is an exact analog of the deterministic gradient procedure:

$$x_{k+1} = x_k - K_k \frac{\partial z(x_k)}{\partial x_k}$$

where the noisy measurement is $z_i = \theta(x_i) + v_i$. This recursion will converge to the solution of $h(x) = 0$ if the gain K meets the following requirements:

$$\begin{aligned} \lim_{k \rightarrow \infty} K_k &= 0 \\ \sum_{k=1}^{\infty} K_k &= \infty \\ \sum_{k=1}^{\infty} K_k^2 &< \infty \end{aligned}$$

Another algorithm for system identification may be described by:

$$\hat{\theta}_{k+1} = \hat{\theta}_k - \left[\frac{\gamma}{k+1} \right] \frac{(z_{k+1}^T \hat{\theta}_k - y_{k+1}) \hat{\theta}_k}{\|\hat{\theta}_k\|}$$

where we have replaced the derivative by a term best defined as a directional error.

7.3. Spline Approximation

The typical use for the spline approximation is to construct a piecewise polynomial to fit data. An exact fit involves interpolation; an approximate fit uses least squares (minimum mean square error) approximation. To explain the structure and advantages of the spline, consider a truncated Taylor series (expanded about x_0 where D^i is the i^{th} derivative):

$$\sum_{i=0}^n \frac{(x - x_0)^i}{i!} D^i f(x_0)$$

This polynomial should provide a satisfactory approximation for $f(x)$ if the function is sufficiently smooth and x is sufficiently close to x_0 . But, if the function must be approximated over a larger interval, the degree of the polynomial may have to be unacceptably large.

The alternative to a higher order polynomial is to subdivide the interval into sufficiently small intervals such that, on each interval, a polynomial with a relatively low degree can provide an adequate approximation.

There are two forms for a spline. The pp-form describes the spline in terms of its breakpoints and the local polynomial coefficients of its pieces. The j^{th} piece of the spline looks like this:

$$p_j(x) = \sum_{i=0}^k \frac{(x - \xi_j)^i}{i!} c_{ij}$$

The B-form of the spline describes a spline as a linear combination and is suited for defining smoothness requirements across breakpoints. Here is the j^{th} B-spline of order k for the knot sequence $t_1 \leq t_2 \leq \dots \leq t_{n+k}$:

$$B_j(x) = \sum_{j=0}^n B_{jk} a_j.$$

B_{jk} is a pp-spline of degree less than k with breakpoints $t_j \leq t_{j+1} \leq \dots \leq t_{j+k}$.

The construction of a spline is a stable and straightforward mathematical procedure [27]. At the breakpoints, derivatives are continuous. At the end points, two conditions are possible. In the "natural" cubic spline, the second derivative is zero. In the "not-a-knot" end condition, the jump in the third derivative is zero.

Once developed, the spline can be evaluated, integrated, differentiated, augmented or cut.

7.4. Canonical Variate Analysis (CVA)

7.4.1. Introduction

The canonical variate method is a prediction error approximation technique that optimally predicts future responses based on a reduced order state space system [28,29]. In the statistical literature, the canonical variate problem is one of maximizing the correlation between two sets of variables. Here we will use the technique to chose nonlinear combinations of past data to predict the future data [30].

Observations coming from the behavior we desire to model are separated into the past $p(t)$ of a vector process and the future $f(t)$ of another vector process. They are assumed to be jointly stationary:

$$\begin{aligned} \mathbf{p}^T &= \left(\mathbf{y}^T(t), \mathbf{y}^T(t-1), \dots, \mathbf{u}^T(t), \mathbf{u}^T(t-1), \dots \right)^T \\ \mathbf{f}^T &= \left(\mathbf{y}^T(t+1), \mathbf{y}^T(t+2), \dots, \mathbf{y}^T(t+l) \right)^T \end{aligned}$$

where the vector process $\mathbf{p}(t)$ can include both inputs and outputs: $\left[\mathbf{y}^T(t), \mathbf{u}^T(t) \right]$.

The optimal k^{th} order linear predictor $\hat{\mathbf{f}}(t)$ of the past is measured by the prediction error:

$$E \left\{ \left\| \mathbf{f} - \hat{\mathbf{f}} \right\|_{\Lambda^{-1}}^2 \right\} \equiv E \left\{ \left(\mathbf{f} - \hat{\mathbf{f}} \right)^T \Lambda^{-1} \left(\mathbf{f} - \hat{\mathbf{f}} \right) \right\}$$

where Λ is arbitrary positive semidefinite, so that Λ^{-1} is a quadratic weighting matrix that is possibly singular. The CVA problem is to determine $\mathbf{c}(t) = \mathbf{J}_k \mathbf{p}(t)$ and $\mathbf{d}(t) = \mathbf{L}_k \mathbf{f}(t)$ (a function of reduced order memory) such that the prediction error is minimized.

The connection between CVA and metamodeling is not direct and much of the literature is very confusing or misleading. First, recall that the metamodel is a reduced order model that is the result of an optimal projection of the higher order model onto a subspace of reduced dimensions. It can be shown that projection operators on a Hilbert Space of nonlinear functions can be expressed as a conditional expectation [28]. It can also be shown that eigenvectors of this conditional expectation have a common eigenvalue which is equal to the squared maximal correlation [30]. If a process has a rational power spectrum (i.e., is a finite order Markov process) then there are a finite number of nonzero canonical correlations between the past and future outputs.

For nonlinear CVA, the Hilbert space for nonlinear functions can be approximated by a finite number of polynomial functions in a given number of lags of the past inputs and outputs of the system. For nonlinear models, it is necessary to include all lower order terms of a power to insure that the model is invariant to a linear transformation of the data.

The solution to the canonical variate problem is expressed by putting the covariance structure of the past and future data in a canonical form such that in this new basis the norm of the weighted prediction error is the sum of squares. This is equivalent to finding \mathbf{J} and \mathbf{L} such that:

$$\begin{aligned} \mathbf{J} \Sigma_{pp} \mathbf{J}^T &= \mathbf{I}_m \quad \mathbf{L} \mathbf{A} \mathbf{L}^T = \mathbf{I}_n \\ \mathbf{J} \Sigma_{pf} \mathbf{L}^T &= \text{Diag} \{ \gamma_1, \geq \gamma_2, \geq \dots, \geq \gamma_q, \geq 0, \dots, 0 \} = \mathbf{D} \end{aligned}$$

where Σ_{pp} , Σ_{ff} , and Σ_{pf} are the covariance matrices of past, future, and cross covariance of the past and future data defined by:

$$\Sigma = \begin{pmatrix} \Sigma_{pp} & \Sigma_{pf} \\ \Sigma_{fp} & \Sigma_{ff} \end{pmatrix}$$

$\text{Diag}\{\gamma_1, \geq \gamma_2, \geq \dots, \geq \gamma_q, \geq 0, \dots, 0\}$ is a diagonal matrix with the singular values on the diagonal. Since the past and future basis in the new basis are orthonormal and uncorrelated, the singular values are also the correlations between the canonical variates p and f .

Determining this structure requires multiple steps. First, given the past and future vectors, the mean is removed to meet the constraints of the alternating conditional expectation (ACE) algorithm that will be used to determine the maximum correlation between transformed input and output variables c and d [31]. Then a (Σ_{pp}, Λ) singular value decomposition of Σ_{pf} will determine a J and L such that after the transformations $c(t) = J_k p(t)$ and $d(t) = L_k f(t)$ the covariances $\Sigma_{cc} = \Sigma_{dd} = I$ [32].

Since we have used a finite data sequence to determine the correlation matrix, Σ_{cd} is not in general diagonal and, consequently, the larger singular values in the sequence $\{\gamma_1, \geq \gamma_2, \geq \dots, \geq \gamma_q, \geq 0, \dots, 0\}$ will be greater than 1. For linear systems, a sequential selection of transformed variables can be made such that the newly selected pair of inputs and outputs are orthogonal to the previously selected set. For nonlinear systems this is not sufficient.

As stated, in a linear system, independent variables are orthogonal. This does not generalize to nonlinear systems, orthogonality is not sufficient to exclude redundancy among variables. For nonlinear systems, stochastic independence is required. The maximal correlation is defined by:

$$\rho(p, f) = \sup_{p, f} \rho(p(y), f(y)) = \sup_{\substack{p, f \\ \|p\|_2 = 1 \\ \|f\|_2 = 1}} E\{p(y), f(y)\}$$

If $\rho(p, f) = 0$, then $p(y), f(y)$ are statistically independent. Therefore, to find the optimal combination of past data to predict the future, we want the maximal correlation.

Maximal correlation is also consistent with the objective to accommodate the finite data sequence and reduce the above joint covariance matrix to the following form:

$$\Sigma = \begin{pmatrix} I_n & D \\ D^T & I_r \end{pmatrix}$$

The tool of canonical correlations is designed to expressly accomplish that task [33]. The ACE algorithm is designed to provide optimal transformations that will minimize the regression error. By doing so, it will also maximally correlate the variates [31]. With maximal correlation, the variables will be statistically independent (as required for nonlinear systems).

The ACE algorithm minimizes $E\left\{\sum_{i=1}^r g(d) - \sum_{i=1}^n f(c)\right\}^2$ by alternating between two minimization problems:

- Given $g(d)$, find functions $f(c)$ to minimize $E\left\{\sum_{i=1}^r g(d) - \sum_{i=1}^n f(c)\right\}^2$.
- Given $f(c)$, find functions $g(d)$ to minimize $E\left\{\sum_{i=1}^r g(d) - \sum_{i=1}^n f(c)\right\}^2$.

Consequently, two conditional expectations are required: $E\{g(d)|c\}$ and $E\{f(c)|d\}$. If $\Sigma_{cc} = \Sigma_{dd} = I$ and if $\Sigma_{cd} = \text{Diag}\{\gamma_1, \gamma_2, \dots, \gamma_q, 0, \dots, 0\}$, i.e. if c and d are maximally correlated, then the following projection relationships hold [34]:

$$E\{g(d)|c\} = \Sigma_{dc} f(c)$$

$$E\{f(c)|d\} = \Sigma_{cd} g(d)$$

These relationships can be used in the ACE algorithm. The outline of the CVA algorithm is shown below. Additional steps must be added to insure proper rank, etc.

7.4.2. CVA Algorithm

1. For the identification problem, let $\Lambda = \Sigma_{ff}$.
2. Determine the number of lags and degree of nonlinearity to be considered. Compute the nonlinear functions of the past data for these lags and degree. Parse the future data. The number of future data points per observation is equal to the number of polynomial functions of the past data. Subtract the mean from each set of input and output variables. Compute the covariance matrices for the newly defined past and future data (Σ_{pp} and Σ_{ff}).
3. Compute the square roots $\Sigma_{pp}^{-1/2} = U_1 S_1^{-1/2} V_1^T$, and $\Lambda^{-1/2} = U_2 S_2^{-1/2} V_2^T$ via the singular value decomposition (SVD) of the matrices $\Sigma_{pp} = U_1 S_1 V_1^T$, and $\Lambda = U_2 S_2 V_2^T$.
4. Form the matrix $M = \Sigma_{pp}^{-1/2} \Sigma_{pf} \Lambda^{-1/2}$ and compute the SVD $M = USV^T$. This is the (Σ_{pp}, Λ) SVD of Σ_{pf} .
5. The canonical variate decomposition is obtained by setting:

$$J = U^T \Sigma_{pp}^{-1/2}, \quad L = V^T \Lambda^{-1/2}, \quad \text{and} \quad D = S$$

6. The generalized canonical (orthonormal) variates are:

$$c = Jp \text{ and } d = Lf$$
7. Compute functions $f(c)$ and $g(d)$ that maximally correlate c & d using the ACE algorithm. Since $\Sigma_{cc} = \Sigma_{dd} = I$, and Σ_{cd} should be as diagonal as possible, recompute the covariance matrices and accomplish another (Σ_{pp}, Λ) SVD of Σ_{cd} after each variable update required by the algorithm.
8. After determining the maximal correlation, compute the general transformation matrix T as the least squares solution of $cT = f(c)$. Compute c by $c = T \times (f(c))$. This will provide the input sequence c that is maximally correlated with the output variate d and best meet the requirement for statistical independence.
9. After this transformation, $\Sigma_{cd} \approx \text{Diag}\{\gamma_1, \gamma_2, \dots, \gamma_q, 0, \dots, 0\}$. The minimized prediction error, expressed in terms of the canonical variates is:

$$\min_{k_k} \left\{ \|d - \hat{d}\|_{\Lambda^{-1}}^2 \right\} = \text{tr}\{\Lambda^{-1} \Sigma_{dd}\} - \{\gamma_1 + \gamma_2 + \dots + \gamma_k\}$$

Since, the optimal projection is obtained when the correlation between the past and future is maximized, the selection of the order of the canonical variables can be made sequentially [35]. The relative degree of correlation is given by $\gamma_1 \geq \gamma_2 \geq \dots \geq \gamma_q$. As the γ_i decrease, less information is included. Consequently, these values give a simple manner for determining how many canonical variables to include in the estimate.

7.5. State Space Reconstruction

Canonical variate analysis results in an optimal prediction of the future states from linear combinations of the past. Given the data from CVA, or any other identification method, we can use these predictions to parameterize a state space system for any order $k < q$ via a least squares regression.

Assume the following state space system:

$$\begin{aligned} x_{t,i+1} &= A(\theta) x(t_i) + B(\theta) u(t_i) + M(\theta) w_d(t_i) \\ y(t_i) &= C(\theta) x(t_i) + D(\theta) u(t_i) + O(\theta) w_d(t_i) + v(t_i) \end{aligned}$$

Define $m_{t,i+1} = J_k p(t_{i+1})$ and $m_t = J_k p(t)$. The state space system above expresses $(x_{t,i+1} \ y_t)$ as a linear combination of $(x_t \ u_t)$. We can replace the predicted value of $x_{t,i+1}$ with $m_{t,i+1}$ and x_t with m_t from the canonical variate decomposition and express $(m_{t,i+1} \ y_t)$ as a linear combination of $(m_t \ u_t)$. With this substitution, all of the data is available for a least squares fit of the two data sets leading to:

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix} = \text{cov} \left\{ \begin{pmatrix} m_{t+1} \\ y_t \end{pmatrix}, \begin{pmatrix} m_t \\ u_t \end{pmatrix} \right\} \text{cov}^{-1} \left\{ \begin{pmatrix} m_t \\ u_t \end{pmatrix}, \begin{pmatrix} m_t \\ u_t \end{pmatrix} \right\}$$

with the prediction error given by:

$$\begin{aligned} S &= \begin{pmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{pmatrix} = \text{cov} \left\{ \begin{pmatrix} m_{t+1} \\ y_t \end{pmatrix}, \begin{pmatrix} m_{t+1} \\ y_t \end{pmatrix} \right\} \\ &\quad - \text{cov} \left\{ \begin{pmatrix} m_{t+1} \\ y_t \end{pmatrix}, \begin{pmatrix} m_t \\ u_t \end{pmatrix} \right\} \text{cov}^{-1} \left\{ \begin{pmatrix} m_t \\ u_t \end{pmatrix}, \begin{pmatrix} m_t \\ u_t \end{pmatrix} \right\} \text{cov} \left\{ \begin{pmatrix} m_t \\ u_t \end{pmatrix}, \begin{pmatrix} m_{t+1} \\ y_t \end{pmatrix} \right\} \end{aligned}$$

so that

$$Q = S_{11}$$

$$O = S_{21} S_{11}^{-1}$$

$$R = S_{22} - S_{21} S_{11}^{-1} S_{12}$$

8. RELATIONSHIPS BETWEEN METHODS

Recall that we have defined a model structure to represent a subset of system behaviors that are of interest. Even if we identify the MPUM, this representation is not unique. Similarly, the methods used to identify model parameters can converge, depending on the model structure assumptions.

8.1. ARX Models

For an unknown mean of a normally distributed (gaussian) ARX system, the maximum-likelihood estimate is the sample mean, the least squares solution, the unbiased estimate, consistent and the best linear unbiased estimate.

8.2. Maximum Likelihood Approaches

Consider the maximum likelihood approaches.

If the process disturbance $w_d(t) = E \left\{ \begin{bmatrix} w_i \\ w_i \end{bmatrix} \begin{bmatrix} w_j^T & w_j^T \end{bmatrix} \right\} = 0$, then the gain of the Kalman filter $K(\theta) = 0$. As a result, the state estimate becomes deterministic and the covariance of the innovations sequence equals the strength of the measurement noise (the only source of error remaining). In this case, the ML estimator is identical to a minimum mean square estimator (sometimes called the output-error method).

Without the effects of measurement noise, $v_d(t) = E \left\{ \begin{bmatrix} v_i \\ v_i \end{bmatrix} \begin{bmatrix} v_j^T & v_j^T \end{bmatrix} \right\} = 0$, the state error equation becomes:

$$\tilde{x}(t_i) = x(t_i) - \hat{x}(t_i) = C^{-1}(\theta) [z_i - D(\theta)u(t_i)] - A(\theta)x(t_{i-1}) - B(\theta)u(t_{i-1})$$

which can be reformulated as a predictor model and minimized using the predictor error method.

Recall that the definition of the predictor error method was given as:

$$\hat{\theta}_N = \hat{\theta}_N(D) = \arg \min_{\theta \in D} \{V(\theta, D)\}$$

and that the likelihood function for the maximum likelihood method was found to be:

$$LLF(\hat{\theta}) = \frac{1}{2} \sum_{i=1}^N \{ \tilde{z}_i^T (P)^{-1} \tilde{z}_i \} + \frac{N}{2} \log \det(P) + \frac{Nm}{2} \log 2\pi$$

If we let $V(\theta, D) = \frac{1}{2} \sum_{i=1}^N \{ \tilde{z}_i^T (P)^{-1} \tilde{z}_i \} + \frac{N}{2} \log \det(P) + \frac{Nm}{2} \log 2\pi$, then we can see that maximum likelihood identification is a special case of the predictor error method. The

reason for the different formulation of the problem is that the calculation of the LLF is quite complicated and often requires a predictor such as a Kalman-Bucy filter.

Obviously, if the prior PDF is not significant, the MAP estimate is similar to the ML estimate.

9. REFERENCES

1. Franklin, Powell, Workman, *Digital Control of Dynamic Systems*, 2nd Edition, Addison-Wesley Publishing Co., Reading MA, 1990.
2. L. Ingber, "Statistical Mechanics of Combat and Extensions," reprint from *Toward a Science of Command, Control, and Communications*, AIAA, December 1993.
3. P.M. Morse, *Thermal Physics*, W.A. Benjamin, NY, 1965.
4. A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, McGraw-Hill, 1965.
5. G. Milne, *State-Space Identification Tool for use with MATLAB: User's Guide*, The MathWorks, Inc, Shereborn, MA, March 1988.
6. L. Ljung, *System Identification: Theory for the User*, Prentice-Hall, New Jersey, 1987.
7. P.S. Maybeck, *Stochastic Models, Estimation, and Control*, Vol 2, Academic Press, 1982.
8. Kendall, Stuart, and Ord, *Kendall's Advanced Theory of Statistics*, Volume 1, Distribution Theory', Oxford University Press, 1987.
9. *Engineering Design Handbook*, U.S. Army Material Command Pamphlet AMCP 706-110, December 1969.
10. D. C. Montgomery, *Design and Analysis of Experiments*, John Wiley & Sons, NY, 1991.
11. D. G Luenberger, *Optimization by Vectro Space Methods*, John Wiley & Sons, NY, 1969.
13. Press, et al, *Numerical Recipes*, Cambridge University Press, 1986.
14. J. Juang and R. S. Pappa, "An Eigensystem Realization Algorithm for Modal Parameter Identification and Model Reduction," *J. Guidance*, vol 8, pp 620-627, September-October 1985.
15. M. A. Norris, "Constrained Eigensystem Realization Algorithm for Lightly Damped Distributed Structures," *J. Guidance*, vol 14, pp 1014-1020, September-October 1991.
16. C. Chen and J. Huang and M. Phan and J. Juang, "Integrated System Identification and State Estimation for Control of Flexible Space Structures," *J. Guidance*, vol 15, no. 1, pp 88-95, January-February 1992.

17. J. E. Cooper and J. R. Wright, "-ESpacecraft In-Orbit Identification Using Eigensystem Realization Methods," *J. Guidance*, vol 15, no. 2, pp 352-359, March-April 1992.
18. J. Juang, M. Phan, L. G. Horta, and R. W. Longman, "Identification of Observer/Kalman Filter Markov Parameters: Theory and Experiments," *J. Guidance*, vol 16, no. 2, pp 320-329, March-April 1993.
19. J. C. Willems, "Paradigms and Puzzles in the Theory of Dynamical Systems," *IEEE Trans. on Automat. Contr.*, vol 36, no. 3, pp 259-294, March 1991.
20. J. P. Norton, *An Introduction to Identification*, Academic Press, San Diego, CA, 1986.
21. R. E. Maine and K.W. Iliff, "Formulation and Implementation of a Practical Algorithm for Parameter Estimation with Process and Measurement Noise," *SIAM J. APPL. Math.*, Vol. 41, No. 3, Decembr 1981.
22. C. R. Rao, *Linear Statistical Inference and its Applications*, Wiley, New York, 1968.
23. Goodwin, Payne, *Dynamic System Identification*, Academic Press, 1977.
24. A. L. Ingber, Draft of "Statistical Mechanical Aids to Calculating Term Structure Models," *J. Phys. Rev.* Vol 42, 1990.
25. L. Ingber, "Simulated Annealing: Practice versus Theory," reprint from *Journal Mathl. Comput. Modeling*, December 1993.
26. Sinha, Kusta, *Modeling and Identification of Dynamic Systems*, Van Nostrand, 1983.
27. *A Practical Guide to Splines*, Applied Math. Sciences Vol. 27, Springer Verlag, New York, 1978.
28. W. E Larimore, "System Identification and Filtering of Nonlinear Controller Markov Processes by Canonical Variate Analysis," Final Report for AF Office of Scientific Research, October 27, 1989.
29. H. Akaike, "Canonical Correlation Analysis of Time Series and the Use of an Information Criterion," *System Identification: Advances and Case Studies*, R.K. Mehra and D.G. Lainiotis, eds. Academic Press, New York, pp 27-96., 1976.
30. W. E. Larimore and J. Baillieul, "Identification and Filtering of Nonlinear Systems Using Canonical Variate Analysis," *Proc. of 29th Conf. on Decision and Control*, pp 635-640, 1990.

31. L. Breiman and J.H. Friedman, "Estimating Optimal Transformations for Multiple Regression and Correlation," *Journal of the American Statistical Association*, Vol. 80, No. 391, pp 580-598, September 1985.
32. H. Hotelling, "Relations Between Two Sets of Variables," *Biometrika*, Vol. 28, pp 321-377, 1936.
33. L. M. Ewerbring and F. T. Luk, Parallel Algorithms for Canonical Variates Computation, *Proc. of 29th Conf. on Decision and Control*, pp 617-628, December 1990.
34. W. E. Larimore, System Identification, Reduced-Order Filtering and Modeling via Canonical Variate Analysis, *Proc. of 29th Conf. on Decision and Control*, pp 445-451, 1990.
35. W. E. Larimore, Canonical Variate Analysis in Identification, Filtering, and Adaptive Control, *Proc. of 29th Conf. on Decision and Control*, pp 596-604, December 1990.

CHAPTER 7

DETERMINATION OF MODEL SET AND ORDER.

1. CHAPTER OUTLINE

1. Chapter Outline	7-1
2. Introduction	7-2
3. General.....	7-3
3.1. Issues in the Selection of the Set and Order.....	7-3
3.2. Equivalent Representations.....	7-4
3.3. Canonical Representations	7-5
3.3.1. Single-Input-Single-Output (SISO) State Space Descriptions	7-5
3.3.2. Multi-Input-Multi-Output (MIMO) State Space Descriptions	7-7
3.3.3. Matrix Fraction Descriptions (MFD)	7-8
3.4. Minimal Realizations, Observability, and Identifiability	7-10
3.4.1. Minimal Realizations	7-10
3.4.2. Observability	7-10
3.4.3. Identifiability	7-11
3.5. Summary of Methods.....	7-11
4. Model Set Selection.....	7-12
5. Generalized Model Order Determination.....	7-15
5.1. Canonical Variate Analysis (CVA)	7-15
5.2. Stochastic Embedding.....	7-15
5.2.1. System Definition	7-17
5.2.2. Estimation of the Noise and Modeling Error.....	7-18
5.2.3. Model Set Selection	7-18
5.3. Eigenstructure Realization Algorithm (ERA).....	7-19
5.4. Residual Error Method	7-19
5.5. Correlation Method	7-20
6. Method Based Model Order Determination	7-21
6.1. Prediction Error Methods	7-21
6.2. Maximum Likelihood Methods	7-21
7. Model Order Reduction Methods	7-23
7.1. Exhaustive search	7-23
7.2. Balanced form.....	7-23
7.3. Optimal Projection.....	7-24
8. References.....	7-28

2. INTRODUCTION

Chapter 5 began the presentation of research to support Objective 2 and covers the models available to support decisions associated with "Step 9: Postulate a metamodel." While Chapter 5 and 6 provided a summary of the model structures available and the identification methods, this chapter continues with methods to select the set and order of the metamodel.

First the chapter provides a general introduction to the problem, some of the potential realizations, and issues associated with the selection of a particular realization. Then we discuss the selection of the model set. This selection is driven by the *a priori* information on the purpose of the metamodel combined with the characteristics of the simulation model itself.

After discussion of the model set, the chapter discusses issues associated with the selection or determination of order from two perspectives. Since the generation of the metamodel may be an iterative process, we may need to revisit decisions when the model is not valid or does not meet the requirements. Therefore, we need two kinds of methods. First, we must make an initial guess as to the model to generate the first metamodel. In this situation, we will concentrate on some nonparametric techniques that are based primarily on the characteristic of the data available. Once we have selected a model and identified (parameterized) a model using one of the methods in Chapter 6, and the metamodel does not meet *a priori* requirements, we may have to re-address the issue of model set and order. In this case, unless we change the model set, we will use methods that are applicable to specific model structures and identification techniques.

Again, the actual procedures for making the selections are included in Chapter 10, Metamodeling Combat Simulations.

3. GENERAL

3.1. Issues in the Selection of the Set and Order

To introduce the selection of the set and order, we first discuss the errors that can occur with an improper selection.

In the identification problem of a truly unknown system, there are three sources of potential error: Modeling Error, Process Noise, and Measurement Noise. Modeling Errors stem from the fact that you really do not have the correct model set and order. Consequently, calculations using this model will not agree with reality. Process noise comes from the fact that, even if we have selected the proper model and (possibly) order, we have not included all of the detail in the model that exists in the real system. In this case, we represent the lack of detail (knowledge) as a stochastic process of some known magnitude. Measurement noise originates from two sources. First, our instruments are not perfect and the output of the instrument contains noise. Second, the fact that we take a measurement results in an interaction with the system that could affect the future system behavior (similar to the Heisenberg uncertainty principle in quantum mechanics).

When you are merely fitting a polynomial to the Input-Output map, there are measures that can be used to determine the order of the operation. You can look at the maximum and average errors, coefficient of determination, p matrix, etc., to determine if the added accuracy obtained by including additional terms is cost effective in terms of the variation induced. These measures make sense, because the input-output map contains the combined effects of the above errors.

If you are trying to identify a process (the plant) that actually generated the input-output map, however, these measures are not directly applicable. The process does not include measurement or modeling errors that are observable in the observations.

Consequently, the problem arises from the fact that it is not possible to isolate the effect of these errors *a priori*. As a simple example, consider the two linear approximations in Figure 7.3.1. The first approximation is constrained by the boundary conditions, while the second is a minimum mean square error (MSE) fit. The accuracy (in terms of the linear parameters) of the first fit is wholly dependent on the combined errors present in the first and last data points. On the other hand, an unrestricted minimum means square fit (that does not begin at the first data point or terminate at the final data point) could contain significant modeling error to compensate for the combined effects of the measurement and process errors. In either case, it is not possible to say which model is better since they contain different error sources. However, even though the MSE fit may not be a better model, it will minimize the combined effects of the errors and result in an approximation that is closer (in the mean) to the actual data.

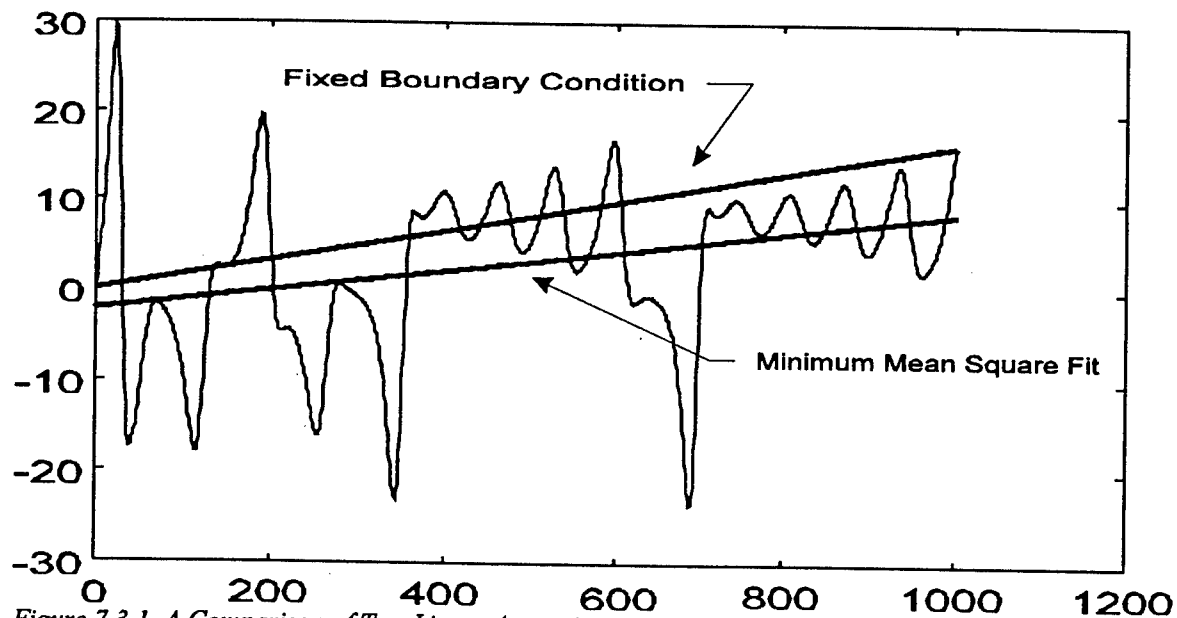


Figure 7.3.1 A Comparison of Two Linear Approximations.

The task, then, is to arrive at a procedure for systematically isolating the errors in the selection of model set and order.

It should be noted that with finite noisy data, the optimal model order is typically smaller than the exact model order, and the quest for the true model order on the basis of finite data is a misguided pursuit [1]. However, if the model orders are overestimated in certain model structures, global and local identifiability will be lost. Our selection of model set and order will be based on the objective of providing a metamodel of sufficient order to maintain identifiability while minimizing bias and variance in the parameter estimates.

The definition of a metamodel combined two elements. The behavior of the system and the representation of that behavior. The representation is defined by the model set and order. Representations of a behavior are not unique and the order of the metamodel is also a function of that choice. This observation leads to a discussion of equivalent representations.

3.2. Equivalent Representations

Recall that we are discussing the representation of allowable behaviors of a system and that multiple representations are available. As stated in Chapter 5, there is a correspondence between the different representations. Consider, for example, a third order ARX model:

$$y(t) + a_1 y(t-1) + a_2 y(t-2) + a_3 y(t-3) = b_1 u(t-1) + b_2 u(t-2) + b_3 u(t-3)$$

This model is equivalent to the Transfer Function model: $G(z, \theta) = \frac{b_1 z^2 + b_2 z + b_3}{z^3 + a_1 z^2 + a_2 z + a_3}$,

where $z = q^{-1}$ is the z transform. The Transfer Function is an external description of the system. It is also equivalent to the following state space model:

$$x_{t_{i+1}} = A(\theta) x(t_i) + B(\theta) u(t_i)$$

$$y(t_i) = H(\theta) x(t_i) + v(t_i)$$

with

$$A(\theta) = \begin{bmatrix} -a_1 - a_2 - a_3 & -b_1 - b_2 - b_3 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad B(\theta) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

$$H(\theta) = [-a_1 \quad -a_2 \quad -a_3 \quad b_1 \quad b_2 \quad b_3]$$

The selection of the model structure will determine the realization. With this selection, there are other impacts. One of these impacts is observability. A realization is observable if $x(t_0)$ can be deduced from knowledge of A , H , and $\{y(t), t_0 \leq t \leq t_f\}$. For example, the state-space system has six states to describe a third-order transfer function (it is called "nonminimal"). With this system representation, there are three states that are unobservable from the output for any values of a_i or b_i . This lack of observability would impact our ability to identify the process. Consequently, we would like to select a structure that results in a minimal realization. To address minimal realizations, however, we must discuss standard or canonical forms that exist within a given model set.

3.3. Canonical Representations

Given a set X and an equivalence relation¹, denoted \sim , on that set, we can decompose the set into subsets each composed of elements that are equivalent to each other. This decomposition results in a quotient set of subsets of X . Given a set X and an equivalence relation \sim , a subset C of X will be a set of canonical forms for X under \sim if for every $x \in X$ there exists one and only one $c \in C$ such that $x \sim c$ [2].

3.3.1. Single-Input-Single-Output (SISO) State Space Descriptions

For scalar linear systems, there are a number of "standard" canonical forms. An irreducible transfer function has four canonical realizations: observer form, controller form, observability form, and the controllability form. Using the same state space structure as above the controller form is:

¹The properties of an equivalence relation are: Transitivity, Symmetry, and Reflexivity

$$A_c(\theta) = \begin{bmatrix} -a_1 & -a_2 & -a_3 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad B_c(\theta) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

$$H_c(\theta) = [h_1 \quad h_2 \quad h_3]$$

while observer form is:

$$A_o(\theta) = \begin{bmatrix} -a_1 & 1 & 0 \\ -a_2 & 0 & 1 \\ -a_3 & 0 & 0 \end{bmatrix} \quad B_o(\theta) = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

$$H_o(\theta) = [1 \quad 0 \quad 0]$$

These forms are "duals" of each other with $A_o = A_c^T$, $B_o = H_c^T$, $H_o = B_c^T$

In the same manner, the controllability form (the name comes from the fact that the controllability matrix - see below- is the identity matrix) is:

$$A_{co}(\theta) = \begin{bmatrix} 0 & 0 & -a_1 \\ 1 & 0 & -a_2 \\ 0 & 1 & -a_3 \end{bmatrix} \quad B_{co}(\theta) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

$$H_{co}(\theta) = [h_1 \quad h_2 \quad h_3] \begin{bmatrix} 1 & a_1 & a_2 \\ 0 & 1 & a_3 \\ 0 & 0 & 1 \end{bmatrix}^{-1}$$

and the observability form (whose controllability matrix is the identity matrix) is:

$$A_{ob}(\theta) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -a_1 & -a_2 & -a_3 \end{bmatrix} \quad B_{ob}(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ a_1 & 1 & 0 \\ a_2 & a_1 & 1 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

$$H_{ob}(\theta) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

with the duality between these forms shown by $A_{ob} = A_{co}^T$, $B_{ob} = H_{co}^T$, $H_{ob} = B_{co}^T$.

These forms can be expanded into larger systems in diagonal or Jordan forms (cf. [2]).

From the above discussion, it is clear that a given system can have many equivalent realizations. Since we can form another realization by a change of variables $x(t) = T\tilde{x}(t)$, $\det\{T\} \neq 0$ with the matrices related by $\tilde{A} = T^{-1}AT$. These matrices are called "similar" and the transformations are referred to as "similarity transformations."

3.3.2. Multi-Input-Multi-Output (MIMO) State Space Descriptions

Analogous to the SISO canonical forms MIMO canonical forms can also be defined. The block controller form is:

$$A_c(\theta) = \begin{bmatrix} -a_1 I_r & -a_2 I_r & \cdots & -a_r I_r \\ I_r & 0 & \cdots & 0 \\ \vdots & \ddots & & \\ 0 & 0 & I_r & 0 \end{bmatrix} \quad B_c(\theta) = \begin{bmatrix} I_r \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$H_c(\theta) = [H_1 \quad H_2 \quad H_3]$$

The block observer form is:

$$A_o(\theta) = \begin{bmatrix} -a_1 I_m & I_m & \cdots & 0 \\ -a_2 I_m & 0 & \ddots & 0 \\ \vdots & \vdots & & I_m \\ -a_r I_m & 0 & 0 & 0 \end{bmatrix} \quad B_o(\theta) = \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_r \end{bmatrix}$$

$$H_o(\theta) = [I_m \quad 0 \quad \cdots \quad 0]$$

Block controllability and block observability forms can also be defined. The block observability form is used extensively in identification:

$$A_{bo}(\theta) = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1n} \\ A_{21} & A_{22} & \cdots & A_{2n} \\ \vdots & \ddots & & \\ A_{n1} & A_{n2} & \cdots & A_{nn} \end{bmatrix}$$

where

$$A_{ii}(\theta) = \begin{bmatrix} 0 & 1 & \cdots & 0 \\ 0 & 0 & \ddots & 0 \\ \vdots & \ddots & & 1 \\ a_{ii}(1) & a_{ii}(2) & \cdots & a_{ii}(n) \end{bmatrix}$$

$$A_{ij}(\theta) = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 0 & 0 & \ddots & 0 \\ \vdots & \ddots & & 0 \\ a_{ij}(1) & a_{ij}(2) & \cdots & a_{ij}(n) \end{bmatrix}$$

MIMO systems are also related by similarity transformations that have the same (external) transfer function description:

$$G(s) = H(sI - A)^{-1}B = \tilde{H}(sI - \tilde{A})^{-1}\tilde{B}$$

The block forms, however, are not of minimal order and there is no general method to obtain the minimal representation. This discrepancy can lead to differences in the behavior of the internal (State Space) and external (Transfer Function) descriptions of a system.

Although a unique realization is not possible, controllability or observability of the realizations can be established. The system is controllable if the following matrix is "full rank":

$$C = [B \quad AB \quad \dots \quad A^{n-1}B]$$

The system is observable if $O^T = [H^T \quad A^T H^T \quad \dots \quad (A^T)^{n-1} H^T]$ is full rank.

If these matrices are not full rank, we can always find a similarity transformation such that the realization $\{A, B, H\}$ has the form:

$$\tilde{A} = \begin{bmatrix} \tilde{A}_o & 0 \\ \tilde{A}_{21} & \tilde{A}_{\bar{o}} \end{bmatrix}, \quad \tilde{B} = \begin{bmatrix} \tilde{B}_o \\ \tilde{B}_{\bar{o}} \end{bmatrix}, \quad \tilde{H}^T = \begin{bmatrix} \tilde{H}^T_o \\ 0 \end{bmatrix}$$

where $\{\tilde{H}_o, \tilde{A}_o\}$ is observable and $\tilde{H}_o(sI - \tilde{A}_o)^{-1} \tilde{B}_o = H(sI - A)^{-1} B$. (Similar results exist for controllability.)

Consequently, to identify a system we may have to partition the behaviors in such a manner so that we can directly measure the observable states. As stated earlier, the selection of the model set and order has an impact on the indentifiability of system and must be explicitly considered in the selection of the model set. This is especially true in MIMO systems where poor selections can lead to degenerate or unstable systems.

3.3.3. Matrix Fraction Descriptions (MFD)

The invariance of the transfer function for similar realizations leads us to write the transfer function for a linear system $H(s)$ as a "matrix fraction":

$$H(s) = N_R(s)D_R^{-1}(s), \text{ with } D_R(s) = d(s)I_r$$

or

$$H(s) = D_L^{-1}(s)N_L(s), \text{ with } D_L(s) = d(s)I_m$$

With $d(s) = s^p + d_1 s^{p-1} + \dots + d_p$ where p is the degree of the polynomial. The first matrix fraction is a right MFD and can be associated with the block controller realization while the left MFD corresponds to the block observer form. The size of $D_L(s)$, and $N_L(s)$ are $m \times m$, and $m \times r$ respectively.

Given a left MFD $y(s) = D_L^{-1}(s)N_L(s)u(s)$ from one of the identification methods, we can realize a state-space form [3]. First, define the partial state which corresponds to a system of coupled differential equations [2]:

$$\xi(s) = D(s)y(s) = N(s)u(s)$$

Determine the highest order derivative of each $\xi_i(s)$, which is equal to the degree of the i th row of $D(s)$ and write

$$D(s) = S_L(s)D_{hr} + \Psi_L(s)D_{lr}$$

where

$$S_L(s) = \text{diag}\{s^{l_1}, i = 1, \dots, m\}$$

and

$$\Psi_L(s) = \begin{bmatrix} s^{l_1-1}, \dots, s, 1 & 0 & \dots & 0 \\ 0 & s^{l_2-1}, \dots, s, 1 & \dots & 0 \\ \vdots & 0 & \ddots & \vdots \\ 0 & 0 & \dots & s^{l_m-1}, \dots, s, 1 \end{bmatrix}$$

Here l_i are the row degrees of $D(s)$, D_{hr} is the highest row degree coefficient matrix of $D(s)$ and D_{lr} accounts for the remaining lower-row-degree terms of $D(s)$.

Assuming the $D(s)$ is row reduced², we obtain the output equation:

$$y(s) = D_{hr}^{-1}S_L^{-1}(s)[\xi(s) - \Psi_L(s)D_{lr}y(s)]$$

and the observer realization

$$\begin{aligned} A_o &= A_o^0 - D_{lp}D_{hp}^{-1}H_o^0 \\ H_o &= D_{hp}^{-1}H_o^0, \quad B_o = N_{lp} \end{aligned}$$

obtained by modifying the core realization (that is both controllable and observable):

$$A_o^0 = \left\{ \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \dots & 0 \\ 0 & 0 & \ddots & \ddots & 0 \\ 0 & 0 & 0 & \ddots & 1 \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix}, 1 \times l_i, i = 1, \dots, m \right\}$$

$$C_o^0 = \text{block diag}\{[1 \ 0 \ \dots \ 0], 1 \times l_i, i = 1, \dots, m\}$$

$$B_o^0 = I_n, \quad n = \sum_1^m l_i = \deg \det \{D(s)\}$$

²If $\deg \det \{D(s)\} < \sum_1^r l_i$ (the sum of the row degrees is less than degree of the determinant) then $D(s)$ is row reduced.

3.4. Minimal Realizations, Observability, and Identifiability

3.4.1. Minimal Realizations

Having the ability to realize a MIMO system in either State-Space or MFD form, we can now discuss minimal realizations. For a number of reasons (primarily relating to the identifiability of the system and the stability of the representation and the propagation of noise) the objective here is to have the lowest order model possible. Model order reduction can take place by constraint in the initial definition of the model or can be the result of the subsequent reduction of a higher order model.

Since general block form State-Space realizations are not minimal, we must use the MFD realization to consider minimal systems.

Relatively prime polynomials, also called coprime polynomials, have no common factors. MFDs have a similar properties. Since they are matrices, however, both left and right divisors must be considered. Given $H(s) = D_L^{-1}(s)N_L(s)$, an infinity of others can be obtained by choosing any nonsingular³ polynomial matrix $W(s)$ such that:

$$\bar{N}(s) = W^{-1}(s)N(s), \text{ and } \bar{D}(s) = W^{-1}(s)D(s)$$

Then $H(s) = D_L^{-1}(s)N_L(s) = \bar{D}_L^{-1}(s)\bar{N}_L(s)$ and $W(s)$ is a left divisor of $N(s)$ and $D(s)$. Therefore, we can reduce the degree of the MFD by removing left (or right) divisors of the numerator and denominator matrices and would get a minimum degree MFD by extracting the greatest common left (or right) divisor (gcd or gcrd). All gcds are related by similarity transforms.

A nonsingular polynomial matrix whose determinant is not a function of s is called unimodular. Two polynomial matrices with the same number of rows are "relatively left prime" or "left coprime" if all of their gcds are unimodular. $N(s)$ and $D(s)$ will be left coprime if and only if $[D(s) \ N(s)]$ has full rank for every s .

3.4.2. Observability

A minimal realization is one that has the smallest-size A matrix for all tripples $\{A, B, H\}$ satisfying $G(s) = H(sI - A)^{-1}B$, a given transfer funtion. Therefore, minimal systems have irreducible transfer functions and are jointly controllable and observable.

Given any left MFD of $H(s)$, we can always obtain an observable state space realization $\{A, B, H\}$ of order $n = \deg \det[D_L(s)]$ and the minimal degree of the left (or right) MFD is the minimal order of any state space realization.

³A polynomial matrix is nonsingular if $\det D(s)$ is not identically zero.

3.4.3. Identifiability

Having discussed minimal realizations we can add to the discussion of identifiability in Chapter 3 with the following result [3].

Theorem I. Consider the model structure M for the general "black box" model.

$$A(t)y(t) = \frac{B(q)}{F(q)}u(t) + \frac{C(q)}{D(q)}e(t)$$

parameterized by θ with the degree of the polynomial n_a, n_b , etc.. The model structure is locally and globally identifiable at $\theta = \theta^*$ if and only if:

1. There is no common factor to all of $z^{n_a}A^*(z)$, $z^{n_b}B^*(z)$, and $z^{n_c}C^*(z)$.
2. There is no common factor to $z^{n_b}B^*(z)$, and $z^{n_f}F^*(z)$.
3. There is no common factor to $z^{n_c}C^*(z)$, and $z^{n_d}D^*(z)$.
4. If $n_a \geq 1$, then also require that there is no common factor to $z^{n_f}F^*(z)$, and $z^{n_d}D^*(z)$.

3.5. Summary of Methods

Table 7.3.1 summarizes methods discussed in this chapter. Many reduction methods reviewed were either special cases of the methods presented here or were not robust. These methods not included were: aggregation; aggregation with only partial realization; singular perturbation methods; optimum approximation; and finally, matching time moments. Reference [4] has a discussion of these techniques.

Table 7.3.1. Model Structure and Order Determination Methods.

AREA	METHOD
Determination of order	Canonical variate analysis
	Stochastic embedding
	Eigenstructure realization algorithm
	Residual error method
	Correlation method
	Final prediction error
	Akaike's information theoretic criterion
Model order reduction	Exhaustive search
	Balanced form
	Optimal projection

4. MODEL SET SELECTION

Selection of the model set includes the description, class, model structure (predictor or probabilistic), form of the identifier, and the criterion of fit. The selection of the metamodel structure is a function of both the simulation we are going to model and of the *a priori* requirements of the metamodel. Given the simulation, there is a tradeoff between fidelity and domain. If we demand high fidelity over a large domain in a single metamodel, the metamodel will have to be more complex. As either the fidelity or the domain decrease, the complexity of the metamodel decreases.

With respect to metamodeling combat simulations, the systems we are trying to identify are complex, nonlinear, time varying discrete event systems. In general, for this case, the predictor function is a nonlinear function of past observations, and there are too many possibilities for unstructured "black box" models.

Fortunately, in this case, we have explicit knowledge of the nature and characteristics of the model. We have the model, the simulation, that applied the system to the inputs to generate the outputs that we are interested in. Given this information, we can build the nonlinearities into the structure of the metamodel and provide the capability to generate a reduced order approximation of the original model. For what appears to be a nonlinear model, we may want to consider whether nonlinear transformations of data will make it easier to fit a linear model. The ACE algorithm can identify these transformations [5].

If the simulation is in the SIMTAX database, we can select the structure most appropriate for its class of simulations. We can follow up the SIMTAX (external) data with any of the internal information that we may have from prior metamodels (possibly for a different purpose) for the simulation. If the internal information does not exist, we can analyze the simulation to determine the internal structure in accordance with the structure provided in Chapter 4.

The fundamental question is whether the process admits a standard "black box" model description or whether a tailor-made model set must be constructed. In general, we want to try simple model sets first. We must insure, however, that the system structure that generated the data set is within the model set we select. If, for example, the data was generated by a system:

$$y(t) = G_0(q)U(t) + H_0(q)e_0(t)$$

and we select the model set

$$\mathcal{M}: \{G(q, \theta), H(q, \theta) | \theta \in D_{\mathcal{M}}\}$$

then the true system belongs to the model set if and only if

$$G(q, \theta_*) = G_0(q) \text{ and } H(q, \theta_*) = H_0(q) \text{ as } \theta^N \Rightarrow \theta_*$$

If we have the case where:

$$G(q, \rho) = G_0(q) \text{ and } H(q, \eta) = H_0(q) \text{ as } \theta^N \Rightarrow \begin{bmatrix} \rho \\ \eta \end{bmatrix}$$

the model set \mathcal{M} does not contain the true system. We will get the closest approximation to the true system, but the estimates are not guaranteed to be unbiased or minimum variance. The performance of the metamodel as a reduced order version of the high fidelity model will suffer. In the second case presented above we would have to select as the model set \mathcal{M} : $\{G(q, \theta) | \theta \in D_{\mathcal{M}}\}$ with an independent parameterization of H (as shown above).

Consequently, there are two competing requirements, the system structure that generated the data set must be within the selected model set, yet the most general "black box" structure:

$$A(t)y(t) = \frac{B(q)}{F(q)}u(t) + \frac{C(q)}{D(q)}e(t)$$

is too general (not identifiable) for most practical purposes.

To see why the most general "black box" structure is too general, consider the loss function:

$$\bar{V}(\theta) = \bar{E} \left\{ \frac{1}{2} \epsilon^2(t, \theta) \right\}$$

where \bar{E} is defined as ensemble averaging (statistical expectation) over the stochastic process and time averaging over deterministic errors. $\bar{V}(\theta)$ becomes the "average value" of the squared residual error. Writing the Fourier transform of the residuals:

$$\Phi_E(\omega, \theta) = \frac{|G_0(e^{j\omega}) - G(e^{j\omega}, \theta)|^2 \Phi_u(\omega) + \Phi_v(\omega)}{|H(e^{j\omega}, \theta)|^2}$$

we see that minimizing this function is a compromise between minimizing the error in the transfer function $|G_0(e^{j\omega}) - G(e^{j\omega}, \theta)|^2$ and fitting $|H(e^{j\omega}, \theta)|^2$ to the error spectrum⁴. Minimization of the loss function results in:

$$\theta_*(D) \approx \arg \min_{\theta \in D_{\mathcal{M}}} \int_{-\pi}^{\pi} |G_0(e^{j\omega}) - G(e^{j\omega}, \theta)|^2 Q(\omega, \theta_*) d\omega$$

⁴The spectrum, or spectral density of a stationary stochastic process is the Fourier transform of its covariance function. Since we deal with non-stationary signals we need a different definition. If a mean and autocorrelation function exist, the signal is quasi-stationary and we define the power spectrum as:

$$\Phi_s(\omega) = \sum_{\tau=-\infty}^{\tau=\infty} R_s(\tau) e^{-j\omega\tau}$$

with

$$Q(\omega, \theta) = \frac{\Phi_u(\omega)}{|H(e^{j\omega}, \theta)|^2}$$

Therefore Q can be thought of as a weighting function that determines the bias distribution. Factors affecting this distribution will be discussed in Chapter 9, Experimental Design.

If we are considering a state space model structure, we cannot simply "fill in" the matrices A , B , C , D , and K with parameters for identification. The input-output description is defined by $3n$ parameters. To obtain identifiable structures, it is natural to seek realizations that involve $3n$ parameters. The observable canonical form or the block observer form is one such parameterization.

5. GENERALIZED MODEL ORDER DETERMINATION

In this section we will consider techniques that are based on the characteristics of the data and can be used to generate an initial guess at the order of the model. Additional methods, other than the particular techniques presented here are available. These methods, however, reduce to spacial cases of these methods.

5.1. Canonical Variate Analysis (CVA)

Our canonical representation will consist of the first maximum set of independent elements within the sequence of predictors [6]. For linear systems, orthogonality was sufficient to insure independence. For nonlinear systems, stochastic independence is required [7]. Analysis of a set of canonical variates can be used to trade off the bias due to low order model versus the additional variability introduced by too high an order. This analysis is concerned with the canonical representation of the correlation between two sets of random variables [6,8].

Model order selection is part of the CVA method discussed in Chapter 6 and procedures to accomplish the following calculations are included there. After the (Σ_{pp}, Λ) SVD of Σ_{pf} , the calculation of $c = Jp$ and $d = Lf$, and the diagonalization of Σ_{cd} by the ACE algorithm, a direct estimate of the process order can be made. After this transformation, $\Sigma_{cd} \approx \text{Diag}\{\gamma_1, \gamma_2, \dots, \gamma_q, 0, \dots, 0\}$, and the minimized prediction error, expressed in terms of the canonical variates, is:

$$\min_{k_k} \left\{ \|d - \hat{d}\|_{\Lambda^{-1}}^2 \right\} = \text{tr}\{\Lambda^{-1} \Sigma_{dd}\} - \{\gamma_1 + \gamma_2 + \dots + \gamma_k\}$$

As the γ_i decrease, less information is included with the addition of each new variable. Consequently, these values give a simple manner for determining how many canonical variables to include in the estimate.

5.2. Stochastic Embedding

Stochastic embedding is an alternative to the hard bound approach to error quantification for control that assigns a distribution to the errors sources [1]. This technique allows noncompact support and less conservative error bounds. Consider the problem of estimating a model of a dynamic system on the basis of observation of an N-point input-output sequence that was generated by the following system:

$$y_k = G_T(q^{-1})u_k + H(q^{-1})e_k$$

with $G_T(q^{-1})$ and $H(q^{-1})$ rational transfer functions, an i.i.d. stochastic disturbance sequence e_k , and a quasi-stationary input sequence u_k that is independent of e_k .

Consider a predictor $\hat{y}_k(\theta) = G(q^{-1}, \theta)u_k$ parameterized by $\theta \in \mathbb{R}^p$ where the prediction model is a member of the model set:

$$\mathcal{M}_p^* = \{G(q^{-1}, \theta) : \theta \in \mathcal{D}_M \subseteq \mathbb{R}^p\}$$

and there exists a smooth mapping \mathcal{M} between $\theta \in \mathcal{D}_M \subseteq \mathbb{R}^p$ and \mathcal{M}_p^* :

$$\mathcal{M} : \theta \rightarrow \{G(q^{-1}, \theta) \in \mathcal{M}_p^*\}$$

If we determine the optimal θ by minimizing a loss function: $\theta^* = \arg \min_{\theta \in \mathcal{D}} \{V(\theta, D)\}$,

then we can examine the total error between the true transfer function and the estimate and decompose it as follows:

$$\begin{aligned} G_T(e^{-j\omega}) - G(e^{-j\omega}, \hat{\theta}_N) &= G_T(e^{-j\omega}) - G(e^{-j\omega}, \theta^*) \\ &\quad + G(e^{-j\omega}, \theta^*) - G(e^{-j\omega}, \hat{\theta}_N) \end{aligned}$$

The first contribution, $G_T(e^{-j\omega}) - G(e^{-j\omega}, \theta^*)$, is the bias error. The second contribution, $G(e^{-j\omega}, \theta^*) - G(e^{-j\omega}, \hat{\theta}_N)$, is the noise or variance error.

In classical identification theory, the bias error is deterministic. In stochastic embedding, we assume it to be a random variable with a defined distribution. We assume that the probability density function (pdf) of $G_\Delta(e^{-j\omega})$ is $f_\Delta(G_\Delta, \beta)$ and that the pdf of the filtered disturbance is $f_v(v_k, \gamma)$. Therefore, given a model set \mathcal{M}_p^* , and some value θ_0 , we can decompose the true transfer function as $G_T(e^{-j\omega}) = G(e^{-j\omega}, \theta_0) + G_\Delta(e^{-j\omega})$ with $G_\Delta(e^{-j\omega})$ a zero mean stochastic process. Assuming that $G_\Delta(e^{-j\omega})$ can be approximated by a finite impulse response (FIR) model of order $L \leq N$:

$$G_\Delta(e^{-j\omega}) = \Pi(e^{-j\omega})\eta$$

where

$$\begin{aligned} \Pi(e^{-j\omega}) &= [e^{-j\omega}, \dots, e^{-jL\omega}] \\ \eta &= [\eta_1, \dots, \eta_L] \end{aligned}$$

If we also assume that the model structure \mathcal{M} is a mapping with a fixed denominator (only the numerator is parameterized by θ), the model becomes:

$$G(e^{-j\omega}, \theta^*) = \Lambda(e^{-j\omega})\theta$$

with

$$\Lambda(e^{-j\omega}) = [\Lambda_1(e^{-j\omega}), \dots, \Lambda_p(e^{-j\omega})].$$

5.2.1. System Definition

The system can now be written in terms of two independent random variables η representing the total error in the transfer function and v , the disturbance sequence:

$$y_k = \phi_k^T \theta_0 + \psi_k^T \eta + v_k$$

with

$$v_k = H(q^{-1})e_k$$

$$\psi_k^T = [u_{k-1}, \dots, u_{k-L}]$$

where θ_0 is defined by $E\{G_T(e^{-j\omega})\} = G(e^{-j\omega}, \theta_0)$ and ϕ_k is a vector containing filtered versions of the output signal.

Given the following definitions:

$$\Phi^T = [\phi_1, \phi_2, \dots, \phi_N]$$

$$Y^T = [y_1, y_2, \dots, y_N]$$

$$V^T = [v_1, v_2, \dots, v_N]$$

then the least square estimate $\hat{\theta}_N = (\Phi^T \Phi)^{-1} \Phi^T Y$ minimizes a minimum mean square

error criterion: $\theta^* = \arg \min_{\theta \in D} \left\{ \frac{1}{N} \sum_{k=1}^N (y_k - \hat{y}_k(\theta))^2 \right\}$.

Since η_k and v_k are independent:

$$\text{cov}\{\hat{\theta}_N - \theta_0\} = (\Phi^T \Phi)^{-1} \Phi^T (\Psi C_\eta \Psi^T + C_v) \Phi (\Phi^T \Phi)^{-1}$$

where

$$C_\eta = E\{\eta \eta^T\}$$

$$C_v = E\{V V^T\}$$

$$\Psi^T = E\{\psi_1, \dots, \psi_N\}$$

With $Q = (\Phi^T \Phi)^{-1} \Phi^T$ and prior definitions, the error in transfer function can be written as:

$$G(e^{-j\omega}) - G(e^{-j\omega}, \hat{\theta}_N) = (\Pi - \Lambda Q \Psi) \eta - \Lambda Q V$$

This expression clearly separates the modeling error $(\Pi - \Lambda Q \Psi) \eta$ and the noise induced error $\Lambda Q V$.

5.2.2. Estimation of the Noise and Modeling Error

Define the N-vector or residuals

$$\begin{aligned}\varepsilon &= Y - \Phi\hat{\theta} \\ &= \left[I - \Phi(\Phi^T\Phi)^{-1}\Phi^T \right] Y = PY\end{aligned}$$

Since $\theta \in \mathbb{R}^p$, the matrix P is of rank p . To obtain a full rank data vector, represent ε in a new coordinate system. Let R be the $N-p$ independent linear combinations of P orthogonal to Φ and define $W \equiv R^T\varepsilon = R^TY = R^T\Psi\eta + R^TV$. (Additional methods of expressing noise model parameters are presented in [9].)

W is the sum of two random vectors whose probability density functions are functions of unknown parameter vectors describing the distributions of η and v . Therefore, we can compute the pdf of W conditioned on the input data vector U and $\xi^T = (\beta^T, \gamma^T)$ by maximizing the likelihood function $\mathcal{L}(W | U, \xi)$ resulting in the desired estimate for the unknown parameters:

$$\hat{\xi} = \arg \max_{\xi} \{ \mathcal{L}(W | U, \xi) \}$$

For example, if we let

$$\begin{aligned}\eta &\sim N(0, C_{\eta}(\beta)) \\ C_{\eta}(\beta) &= \text{diag} \left\{ \alpha\lambda^k \right\}_{1 \leq k \leq L}\end{aligned}$$

and

$$v_k \sim N(0, \gamma^2)$$

the log likelihood function for the observed data is:

$$l((W | U, \xi)) = -\frac{1}{2} \ln \det[\Sigma] - \frac{1}{2} W^T \Sigma^{-1} W + \text{const} \tan t$$

where

$$\begin{aligned}\Sigma &= R^T\Psi C_{\eta}(\alpha, \lambda) \Psi^T R + \gamma^2 R^T R \\ C_{\eta}(\alpha, \lambda) &= \text{diag} \{ \alpha\lambda, \alpha\lambda^2, \dots, \alpha\lambda^L \}\end{aligned}$$

5.2.3. Model Set Selection

Given the quantified error bounds in the form of the ensemble mean square error:

$$\hat{V}_i(\omega) = E \left\{ \left| G_T(e^{-j\omega}) - G(e^{-j\omega}, \hat{\theta}_N) \right|^2 \right\}$$

and assuming that $\{v_k\}$ is a white noise sequence of variance σ_v^2 , we can select from among different model orders p by considering the generalized information criterion (GIC):

$$\text{GIC}(p) = \hat{\sigma}_v^2 + \frac{p}{N} \hat{\sigma}_v^2 + \frac{1}{2\pi} \int_{-\pi}^{\pi} (\Pi - \Lambda Q \Psi) C_n(\beta_p) (\Pi - \Lambda Q \Psi)^* S_u(\omega) d\omega$$

The three terms are estimates of the effect on the prediction error of: (1) the variance of the noise realization, (2) the parameter errors due to noise in identification, and (3) the modeling error.

5.3. Eigenstructure Realization Algorithm (ERA)

Just as CVA leads to a natural ordering of the variables, the eigenstructure realization algorithm (ERA) also leads directly to the model order. After we have extracted the system Markov parameters from the observer (cf Chapter 6), we recover the state space model by the ERA by defining the following $r \times s$ block data matrix:

$$H(\tau) = \begin{bmatrix} Y_\tau & Y_{\tau+1} & Y_{\tau+2} & \cdots & Y_{\tau+s-1} \\ Y_{\tau+1} & Y_{\tau+2} & Y_{\tau+3} & \cdots & Y_{\tau+s} \\ Y_{\tau+2} & Y_{\tau+3} & Y_{\tau+4} & \cdots & Y_{\tau+s+1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ Y_{\tau+r-1} & Y_{\tau+r} & Y_{\tau+r+1} & \cdots & Y_{\tau+r+s-2} \end{bmatrix}$$

Just as in the CVA method, the order of the system is determined by the singular value decomposition of $H(0)$,

$$H(0) = U \Sigma V^T = U_1 S_1 V_1^T$$

where Σ are all of the singular values. S_1 is an $n \times n$ diagonal matrix of positive singular values that are retained and n will become the order of the system:

5.4. Residual Error Method

This method attempts to determine the observability subindices of a system represented by the block observability form with the order of the system defined as [10]:

$$n = \sum_{i=1}^m n_i$$

with m being the number of outputs. The observability subindices are the dimensions of the individual block observability matrices as defined above.

6. METHOD BASED MODEL ORDER DETERMINATION

In this section we will consider techniques that do not have general applicability and are applicable to specific methods only. These methods are generally used to refine the order of a previously parameterized model.

6.1. Prediction Error Methods

6.1.1. Final Prediction Error

The final prediction error (FPE) provides a solution to the problem of model order determination for an auto-regressive model when least squares estimates are used [6].

For single-output systems the FPE is the one-step ahead prediction error and is defined as [3]:

$$\text{FPE} = \frac{1 + \frac{n}{N}}{1 - \frac{n}{N}} V$$

where n is the number of parameters to be estimated, N is the length of the data record, and V is the loss function. The loss function depends on the criterion of fit (Chapter 6) and ranges from quadratic or robust norms to the maximum log likelihood function (LLF). For multi-output systems, the loss function is defined as the determinant of the estimated covariance matrix of the innovations.

6.2. Maximum Likelihood Methods

6.2.1. Akaike's Information Theoretic Criterion (AIC)

A model structure \mathcal{M} is defined as a differentiable mapping from a connected open subset \mathcal{D}_m of \mathbb{R}^d to a model set $\mathcal{M}(\theta)$, such that the gradients of the predictor functions are stable. This mapping can also be represented as the PDF for the observations:

$$f(\theta; z_1, z_2, \dots, z_N) = f_z(\theta; Z^N)$$

Let $f_m(\theta; N, Z^N)$ be the assumed model for the N observations Z^N . Assume that the true PDF is represented by $f_*(\theta_*; N, Z^N)$. The difference between the two can be measured in terms of the Kullback-Leibler information distance [11]:

$$I(f_*; f_m) = \int f_*(\theta_*; N, x^N) \log \frac{f_*(\theta_*; N, x^N)}{f_m(\theta; N, x^N)} dx^N$$

This distance is also the negative entropy of $f_*(\theta; N, Z^N)$ with respect to $f_m(\theta; N, Z^N)$:

$$S(f_*; f_m) = -I(f_*; f_m)$$

Therefore, we can look for a model that maximizes the entropy with respect to the true system or minimize the information distance to the true system (cf. Appendix 3).

The information measure can be split into two terms (via the log) only one of which is a function of θ . This term, however, requires an expectation with respect to the true system which is not computable. Since we are working with the joint PDF (vice the CPDF) we can replace the expectation with respect to the true system with an estimate that is the LLF from the MLE. Therefore: $I(f_{\cdot}; f_m) \approx -\log f_m(\theta; N, Z^N)$. Since this is a random variable, we can take the average information distance $E_{\hat{\theta}_N} \{I(f_{\cdot}(\theta_{\cdot}; N, Z^N); f_m(\hat{\theta}_N; N, Z^N))\}$ and minimize this function with respect to $\hat{\theta}_N$. An unbiased estimate of this expectation is

$$-\log f_m(\hat{\theta}_N; N, Z^N) - \dim \theta$$

Therefore, substituting this estimate in to the minimization results in:

$$\hat{\theta}_N = \arg \min_{\theta} [-\log f_m(\hat{\theta}_N; N, Z^N) - \dim \theta]$$

This leads to Akaike's information theoretic criterion (AIC).

$$AIC = -2 \max_{\theta} \left\{ \log(f_m(\hat{\theta}_N; N, Z^N)) \right\} + 2 \dim \theta$$

Although the discussion here was in terms of selecting model order, it can also aid in the selection of the model set because the minimization can be performed with respect to the different structures.

This criterion will also favor models with smaller order. If two models are equally likely, then the one with the fewer parameters is chosen [10]

7. MODEL ORDER REDUCTION METHODS

In this situation we already have a model that adequately describes the behavior that we are interested in. As stated above, the objective is to have the lowest order model possible. In this subsection we will consider order reduction of a higher order model. The objective is to insure that all of the terms of the model contribute sufficiently to the reduction of the residual error.

There are two major approaches for model reduction [12]. The first approach uses optimality conditions and performs an exhaustive search for an optimal reduced order model. The second approach transforms the system into a "balanced" form where states are arranged in order of importance.

7.1. Exhaustive Search

Since we are interested in reducing the order, the assumption is that the identified model is of very high order. Consequently, due to the complexity of the representations, the only way of conducting an exhaustive search is by using an efficient technique such as adaptive simulated annealing. In this case, the validity conditions are included in the objective function. Beginning with the identified model and order (this should insure proper initialization), successive lower order approximations of the systems are computed until the desired validity measures cannot be met.

7.2. Balanced Form

A gramian matrix, $G = [G_{ij}]$, with $G_{ij} = \int_{t_0}^{t_f} l_i(\tau) l_j(\tau) d\tau$ is another general method of testing linear independence [2]. Suppose we have a realization:

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{F}\mathbf{x}(t) + \mathbf{G}\mathbf{u}(t) + \mathbf{L}\mathbf{w}(t) \\ \mathbf{y}(t) &= \mathbf{H}(\theta) \mathbf{x}(t) + \mathbf{N}(\theta) \mathbf{u}(t) + \mathbf{v}(t)\end{aligned}$$

Such a realization is observable if $\mathbf{x}(t_0)$ can be deduced from knowledge of \mathbf{F} , \mathbf{H} , and $\{\mathbf{y}(t), t_0 \leq t \leq t_f\}$. This will be true if and only if the columns of $\mathbf{H}(\cdot)\Phi(\cdot, t_0)$, with $\Phi(\cdot, t_0)$ the transition matrix, are linearly independent. Therefore a realization will be observable if and only if the observability gramian:

$$\mathbf{O}(t_0, t_f) = \int_{t_0}^{t_f} \Phi^T(\tau, t_0) \mathbf{H}^T(\tau) \mathbf{H}(\tau) \Phi(\tau, t_0) d\tau$$

is nonsingular.

Consider a discrete system:

$$\begin{aligned}\mathbf{x}_{t_{i+1}} &= \mathbf{A}(\theta) \mathbf{x}(t_i) + \mathbf{B}(\theta) \mathbf{u}(t_i) \\ &\quad + \mathbf{M}(\theta) \mathbf{w}_d(t_i)\end{aligned}$$

$$y(t_i) = C(\theta) x(t_i) + D(\theta)u(t_i) \\ + O(\theta)w_d(t_i) + v(t_i)$$

Transformation of the into a balanced form renders both the controllability W_c and the observability gramians W_o equal and diagonal [12,13].

$$W_c = \sum_{i=0}^{p-1} A^i B B^T (A^T)^i$$

$$W_o = \sum_{i=0}^{p-1} A^i C^T C (A^T)^i$$

Decomposing the grammians using Cholesky decomposition into:

$$W_c = P P^T \quad W_o = Q^T Q$$

and forming the matrix $H = QP$ and performing a SVD gives $H = U \Gamma^2 V^T$.

Defining $R = P V \Gamma^{-1} = Q^{-1} U^T$ and $R^{-1} = \Gamma^{-1} U^T Q = \Gamma V^T P^{-1}$, the balanced form is obtained as:

$$A_b = R^{-1} A R, \quad B_b = R^{-1} B, \quad C_b = C R$$

where the observability and controllability grammians are equal: $W_{cb} = W_{ob} = \Gamma^2$.

If we rearrange the system by states retained, $x_r(i)$, and states truncated, $x_t(i)$, and then define an objective function in terms of the error in the system response using only the retained states:

$$J = \sum_{i=0}^{N-1} [y(\tau) - y_r(\tau)]^T [y(\tau) - y_r(\tau)] \\ = \sum_{i=0}^{N-1} [y_t^T(\tau) y_t(\tau)]$$

we can write that objective function as $J = \text{tr}\{C_t^T C_t \Gamma^2\}$ and minimize the objective function by truncating states corresponding to small diagonal elements of $C_t^T C_t \Gamma^2$.

7.3. Optimal Projection

In general, metamodeling can be considered as an optimal projection onto a reduced order subspace. This method actually uses this projection to reduce the order of a previously identified model. The optimal projection algorithm presented here allows for a frequency weighted quadratic criterion [14].

Consider an n^{th} order, time invariant asymptotically stable system:

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{F}\mathbf{x}(t) + \mathbf{G}\mathbf{u}(t) \\ \mathbf{y}(t) &= \mathbf{H}(\theta) \mathbf{x}(t) + \mathbf{N}(\theta) \mathbf{u}(t)\end{aligned}$$

and find a reduced order system

$$\begin{aligned}\dot{\mathbf{x}}_r &= \mathbf{F}_r \mathbf{x}_r(t) + \mathbf{G}_r \mathbf{u}(t) \\ \mathbf{y}_r(t) &= \mathbf{H}_r(\theta) \mathbf{x}_r(t) + \mathbf{N}_r(\theta) \mathbf{u}(t)\end{aligned}$$

which minimizes the frequency weighted quadratic criterion:

$$J = \int_{-\infty}^{\infty} \left\| \mathbf{R}^{1/2} [\mathbf{G}(j\omega) - \mathbf{G}_r(j\omega)] \mathbf{W}(j\omega) \right\|_F^2 d\omega$$

where $\|\cdot\|_F$ is the Frobenius norm defined as $\|\mathbf{A}\|_F = \left(\text{tr}(\mathbf{A}^T \mathbf{A}) \right)^{1/2}$ with:

$$\mathbf{G}(s) = \mathbf{H}(s\mathbf{I} - \mathbf{F})^{-1} \mathbf{G} + \mathbf{N} \text{ and } \mathbf{G}_r(s) = \mathbf{H}_r(s\mathbf{I} - \mathbf{F}_r)^{-1} \mathbf{G}_r + \mathbf{N}_r$$

If we realize $\mathbf{W}(s)$ as a state space system driven by white noise $\eta(t)$:

$$\begin{aligned}\dot{\mathbf{x}}_w &= \mathbf{F}_w \mathbf{x}_w(t) + \mathbf{G}_w \eta(t) \\ \mathbf{y}_w(t) &= \mathbf{H}_w(\theta) \mathbf{x}_w(t) + \mathbf{N}_w(\theta) \eta(t)\end{aligned}$$

and construct an augmented state $\tilde{\mathbf{x}} = [\mathbf{x}^T \mathbf{x}_w^T \mathbf{x}_r^T]^T$ we can partition the new system as:

$$\dot{\tilde{\mathbf{x}}} = \begin{bmatrix} \mathbf{F}_1 & 0 \\ \mathbf{G}_{r1}\mathbf{H}_{21} + \mathbf{G}_{r2}\mathbf{H}_{22} & \mathbf{F}_r \end{bmatrix} \tilde{\mathbf{x}} + \begin{bmatrix} \mathbf{G}_1 \\ \mathbf{G}_{r1}\mathbf{N}_{w1} \end{bmatrix} \eta$$

$$\boldsymbol{\varepsilon} = [\mathbf{H}_1 - \mathbf{N}_{r2}\mathbf{H}_{22} \quad -\mathbf{H}_r]$$

where the rank $\mathbf{D}_w = p_1 \leq p$ and:

$$\begin{aligned}\mathbf{N} &= [\mathbf{N}_1 \quad \mathbf{N}_2] & \mathbf{G}_r &= [\mathbf{G}_{r1} \quad \mathbf{G}_{r2}] \\ \mathbf{N}_r &= [\mathbf{N}_{r1} \quad \mathbf{N}_{r2}] & \mathbf{H}_w &= \begin{bmatrix} \mathbf{H}_{w1} \\ \mathbf{H}_{w2} \end{bmatrix} \\ \mathbf{F}_1 &= \begin{bmatrix} \mathbf{F} & \mathbf{G}\mathbf{H}_w \\ 0 & \mathbf{F}_w \end{bmatrix} & \mathbf{G}_1 &= \begin{bmatrix} \mathbf{G}\mathbf{N}_w \\ \mathbf{G}_w \end{bmatrix} \\ \mathbf{H}_{21} &= [0_{p1 \times n} \quad \mathbf{H}_{w1}] & \mathbf{H}_{22} &= [0_{(p-p1) \times n} \quad \mathbf{H}_{w2}] \\ \mathbf{H}_1 &= [\mathbf{H} \quad \mathbf{N}_2\mathbf{H}_{w2}] & \mathbf{H}_{11} &= [\mathbf{H} \quad 0_{m \times n_w}]\end{aligned}$$

If we assume that \mathbf{F}_r is asymptotically stable, that $(\mathbf{F}_r, \mathbf{G}_r)$ is controllable, and that $(\mathbf{F}_r, \mathbf{H}_r)$ is observable, the steady state covariance $\tilde{\mathbf{Q}} = \lim_{t \rightarrow \infty} \mathbf{E}[\tilde{\mathbf{x}}(t)\tilde{\mathbf{x}}^T(t)]$ is given by the Lyapunov equation:

$$\tilde{F}\tilde{Q} + \tilde{Q}\tilde{F}^T + \tilde{G}\tilde{G}^T = 0$$

and we can write the frequency weighted quadratic criterion as

$$J = \text{tr}\{\tilde{Q}\tilde{H}^T R \tilde{H}\}$$

Then, the optimal reduced-order model is determined by:

$$\begin{aligned} F_r &= \Gamma(F_1 - Q_a \Sigma_w H_{21}) v_* G(s)^T \\ G_r &= \Gamma[Q_a \Sigma_w (F_1 - Q_a \Sigma_w H_{21}) H_{22}^T] \\ H_r &= H_1 v_* G^T(s) \\ N_r &= N + \begin{bmatrix} 0_{m \times p_1} & H_{11} H_{22}^T \end{bmatrix} \end{aligned}$$

where

$$\begin{aligned} \tilde{Q} &= \begin{bmatrix} Q_1 & Q_{12} \\ Q_{12}^T & A_2 \end{bmatrix} & Q_1 &\in \mathbb{R}^{(n+n_w) \times (n+n_w)} \\ & & Q_2 &\in \mathbb{R}^{n_r \times n_r} \\ \hat{Q} &= Q_{12} Q_2^{-1} Q_{12}^T & Q &\equiv Q_1 - \hat{Q} \\ \Sigma_w &= (N_{w1} N_{w1}^T)^{-1} & Q_a &= Q H_{21}^T + G_1 N_w^T \\ H_{22}^* &= Q H_{22}^T (H_{22} Q H_{22}^T)^{-1} & v &= H_{22}^* H_{22}, v_* = I_{n+n_w} - v \end{aligned}$$

and Q , \hat{Q} , and \hat{P} satisfy:

$$\begin{aligned} & [F_1 - \tau(F_1 - Q_a \Sigma_w H_{21}) v] Q \\ & + Q [F_1 - \tau(F_1 - Q_a \Sigma_w H_{21}) v]^T \\ & - Q_a \Sigma_w Q_a + \tau_* Q_a \Sigma_w Q_a \tau_*^T + G_1 G_1^T = 0 \end{aligned} \quad (1)$$

$$\begin{aligned} & F_1 \hat{Q} + \hat{Q} F_1^T + \tau(F_1 - Q_a \Sigma_w H_{21}) v Q \\ & + Q v^T (F_1 - Q_a \Sigma_w H_{21})^T \tau^T + Q_a \Sigma_w Q_a^T \\ & - \tau_* Q_a \Sigma_w Q_a^T \tau_*^T = 0 \end{aligned} \quad (2)$$

$$\begin{aligned} & \hat{P} (F_1 - Q_a \Sigma_w H_{21}) v_* + v_*^T (F_1 - Q_a \Sigma_w H_{21})^T \hat{P} \\ & + v_*^T H_1^T R H_1 v_* - \tau_*^T v_*^T H_1^T R H_1 v_* \tau_* = 0 \end{aligned} \quad (3)$$

$$\text{rank } \hat{Q} = \text{rank } \hat{P} = \text{rank } \hat{Q}\hat{P} = n_r$$

The solution for Q , \hat{Q} , and \hat{P} can be found by the following algorithm:

1. Initialize $Q^{(0)}$, $v^{(0)}$, and $\tau^{(0)}$.
2. Solve equation 2 for Q as a regular Riccati equation where the terms $\tau(F_1 - Q_s \Sigma_w H_{21})v$, and $\tau \cdot Q_s \Sigma_w Q_s^T \tau^T$ are evaluated from the previous iteration.
3. Solve (2) and (3) for \hat{Q} and \hat{P} .
4. Update $v^{(i)}$, and $\tau^{(i)}$.
5. If Q , \hat{Q} , and \hat{P} did not converge, go to step 2.

8. REFERENCES

1. G. C. Goodwin, M. Gevers, and Brett Ninness, "Quantifying the Error in Estimated Transfer Functions with Application to Model Order Selection," *IEEE Trans. on Automat. Contr.*, vol 37, no. 7, pp 913-928, July 1992.
2. T. Kailath, *Linear Systems*, Prentice-Hall, New Jersey, 1980.
3. L. Ljung, *System Identification Tool for use with MATLAB: User's Guide*, The MathWorks, Inc, Shereborn, MA, July 1991.
4. J. P. Norton, *An Introduction to and Identification*, Academic Press, 1988.
5. L. Breiman and J.H. Friedman, Estimating Optimal Transformations for Multiple Regression and Correlation," *Journal of the American Statistical Association*, Vol. 80, No. 391, pp 580-598, September 1985.
6. H. Akaike, "Canonical Correlation Analysis of Time Series and the Use of an Information Criterion," *System Identification: Advances and Case Studies*, R.K. Mehra and D.G. Lainiotis, eds. Academic Press, New York, pp 27-96., 1976.
7. W. E Larimore, "System Identification and Filtering of Nonlinear Controller Markov Processes by Canonical Variate Analysis," Final Report for AF Office of Scientific Research, October 27, 1989.
8. B. Thompson, *Canonical Correlation Analysis*, Sage Publications, Beverly Hills, CA, 1984.
9. J. Schoukens and R. Pintelon, "Quantifying Model Errors of Identified Transfer Functions," *IEEE Trans. on Automat. Contr.*, vol 39, no. 8, pp 1733-1737, August 1994.
10. Sinha, Kusta, *Modeling and Identification of Dynamic Systems*, Van Nostrand, 1983.
11. S. Kullback, *Information Theory and Statistics*, Dover, 1959.
12. L.G.Horta, J Juang, and R.W. Longmont, "Discrete-Time Model Reduction in Limited Frequency Ranges," *Journal of Guidance, Control, and Dynamics*, Vol. 16, No. 6, November-December 1993.
13. L. Maciejowski, "Multivariable Feedback Control," Addison Wesley, Workingham England, 1989.
14. Y. Halevi, "Frequency Weighted Model Reduction via Optimal Projections," *IEEE Trans. on Automat. Contr.*, vol 37, no. 10, pp 1537-1542, Oct 1992.

CHAPTER 8

DETERMINATION OF MODEL VALIDITY

1. CHAPTER OUTLINE

1. Chapter Outline	8-1
2. Introduction.....	8-2
3. General.....	8-3
3.1. Definitions	8-3
3.2. Error Sources, Components, and Validity Assessments	8-3
4. Measures of Local Validity	8-5
4.1. Parameter and Transfer Function Properties	8-5
4.1.1. Bias and Variance.....	8-5
4.1.2. Validity Measures.....	8-7
4.2. Properties of the Identification Method	8-8
4.2.1. Least Squares Estimator	8-8
4.2.2. Weighted Least Squares Estimator	8-8
4.2.3. General Prediction Error Method (PEM)	8-9
4.2.4. Instrumental Variable (IV) Method.....	8-9
4.2.5. Asymptotic Properties of Maximum Likelihood (ML) Estimators	8-9
5. Measures of Global Validity.....	8-11
5.1. Information Based Model Accuracy Methods.....	8-12
5.1.1. Akaike's Information Theoretic Criterion (AIC)	8-12
5.1.2. Akaike's Final Prediction Error (FPE).....	8-12
5.1.3. Expected Relative Mutual Information.....	8-12
5.2. Data Accuracy Model Validation Methods.....	8-13
5.2.1. Maximum Absolute Error	8-13
5.2.2. Average Absolute Error.....	8-13
5.2.3. Average Absolute Relative Error	8-13
5.2.4. Squared Coefficient of Determination	8-13
5.2.5. Residual Analysis.....	8-13
5.2.5.1. Whiteness Test.....	8-15
5.2.5.2. Independence Between Residuals and Past Inputs.....	8-15
5.2.5.3. Chi-Squared Goodness of Fit	8-16
5.2.5.4. Kolmogoroff-Smirnoff (K-S) Goodness of Fit Test.....	8-16
5.2.6. Lack-of-Fit Test	8-17
5.2.7. Goodness of Fit Test	8-18
5.2.8. Analysis of Variance.....	8-19
5.2.9. Hypothesis Testing Regression Coefficients	8-19
6. Block Structured, Norm-Bounded Uncertainty.....	8-21
7. References.....	8-22

2. INTRODUCTION

Chapters 5,6, and 7 addressed many of the elements that must be considered to generate the metamodel. This chapter follows this discussion and concentrates on methods to accomplish **"Step 13: Access the Validity of the Model."**

This Chapter is closely tied with Chapter 7, "Determination of Model Structure and Order." Recall that when the metamodel is determined, it is not possible to ask "What is the probability that a particular set of fitted parameters is correct?" There is no statistical universe of models from which the correct one is chosen [1]. The validity of the metamodel is highly dependent on the selection of the proper model structure and order. Consequently, some of the measures to determine the validity of the metamodel (e.g., residual error and correlation analysis) when applied to data, can also be used for the initial selection of the structure and order of the model. Also, results of the validity tests are indicative of a proper or improper structure or order.

A good model should: (1) fit the data accurately, (2) be theoretically consistent, and (3) have parameters that have physical meaning and can be measured independently of each other. In addition, a good model should prove useful in making predictions. These are difficult characteristics to quantify.

There are two elements of validity that must be addressed. First, the model must be a consistent estimator of the data that was used to generate the metamodel. This is the aspect of validity that is considered in this chapter. Beyond this, however, there is another aspect of validity that also must be addressed.

The second aspect comes from the fact that metamodeling can be thought of as reduced order modeling where the metamodel is a reduced order model of a high fidelity model. Using the space spanned by the original model as the full order model, the metamodel is a reduced order approximation where this reduced order model is defined over a subspace that is generated by projecting the original space onto this subspace.

Projection operators, however, may not always converge [2]. Consequently, in addition to validation with respect to the data, reduced order models must be explicitly verified for the conditions that will be used. This verification is part of the verification, validation, and accreditation process. This validation will be addressed via procedures in Chapter 10, Metamodeling Combat Simulations.

3. GENERAL

3.1. Definitions

The two elements of validity mentioned above have been formalized in the following definitions:

Verification. The verification process confirms that the model functions as it was originally conceived, specified, and designed. Here we compare the output of the model to the conceptual description, specifications, or definitions that were used in its development.

Validation. Validation addresses the credibility of the model in its depiction of the modeled world. In this case the model is not compared to the structure from which it is developed, but to the phenomenon that it is supposed to represent.

Measuring either of these elements is complex and subject to interpretation. Our working definition of validity will be the lack of error in the estimate of the parameters.

3.2. Error Sources, Components, and Validity Assessments

There are a number of sources of error. The model class, type, or structure could be totally inappropriate for the system being identified. The model order could be incorrect. If the chosen set of models is too small to accommodate the true system, the limit model will deviate from the actual system. Certain elements of the structure may not have been identifiable given the input used for the identification. The identification of the coefficients of the model may not have been accurate enough. Table 8.3.1 outlines some of the errors and their sources.

Errors in estimated transfer functions have two components [3]. The first component, often called variance error, is caused by the noise in the data used for the identification. It usually decreases with increasing record length or excitation. The second component, bias error, is caused by the fact that the parameterized model structure is, at best, a simplified (low order) version of the true system and is typically unaffected by record length, etc. A large, flexible, and well adapted model set results in small bias.

Table 8.3.1. Error Sources.

ERROR	SOURCE	REMARK
Modeling error	Class Description Structure Order	This is the universe from which the most powerful unfalsified model will be chosen.
Identifiability	Input Incorrect order	The input must be persistently exciting. The system parameters must be observable from the output.
Accuracy	Poorly conditioned data	Caused by sampling too fast or too slow. Unmeasurable disturbances. Excessive measurement noise.
Experimental error	Experimental design	Inability to determine y exactly Contains the effects of all unmodeled parameters
Lack of fit	Experimental design	Additional factors are functions of the included parameters

In the case of exact model structure, the Crámer-Rao lower bound produces reasonable estimates of the minimum variance error. In the case of reduced order models, however, the parameters of the model may have no meaning and the classical Crámer-Rao bound does not apply.

If available, estimation of the bias error can be computed by reference to the higher-order model. If there is no noise in the data, one can estimate as many parameters as there are data points if there is sufficient excitation. Characterization of the bias error in the case of a finite set of noisy data is much more difficult.

Model validation is the heart of the identification problem. However, there is no prescribed technique for approaching it. We will assess model validity from both a local and global perspective. Local validity is required for but does not guarantee global validity. Local validity concentrates on the properties of the parameters with respect to the realized system. Global validity includes more general issues as to the domain and range of the model and how well the model fit the observed data.

Many of the methods and measures presented in this chapter can be incorporated into a cross-validation scheme. This technique splits the data into two segments. The first segment is used to generate the metamodel. The second segment is used to validate the metamodel. In this manner, we have a new set of data that provides the truth model.

4. MEASURES OF LOCAL VALIDITY

Local validity concentrates on internal measures. There are two types of validity measures considered here. The first local validity type, parameter properties, are measures of the properties of the parameters themselves. These consist of bias, variance, consistency, and efficiency.

The second type of internal measure consists of properties of the identification method. These properties are characteristic of the criterion and identification method that was used to parameterize the model. Given that the error criterion was minimum mean square error, what was the mean square error? How does this error compare to the theoretically obtainable value? All of the methods are optimal under certain assumptions. We are really measuring how well the data met the assumptions contained in the method.

4.1. Parameter and Transfer Function Properties

4.1.1. Bias and Variance

Bias Error. If $\hat{\theta}$ is an estimate of θ_0 . The bias in the estimate is the difference between the mean value of $\hat{\theta}$ and the true value θ_0 . An estimator where the bias is zero for all N , is called unbiased. From Chapter 7 we used the loss function:

$$\bar{V}(\theta) = \bar{E} \left\{ \frac{1}{2} \varepsilon^2(t, \theta) \right\}$$

where \bar{E} is defined as ensemble averaging (statistical expectation) over the stochastic process and time averaging over deterministic errors. $\bar{V}(\theta)$ becomes the "average value" of the squared residual error. Writing the Fourier transform of the solution to the minimization of the loss function resulted in:

$$\theta_*(D) \approx \arg \min_{\theta \in D_M} \int_{-\pi}^{\pi} |G_0(e^{j\omega}) - G(e^{j\omega}, \theta)|^2 Q(\omega, \theta_*) d\omega$$

with

$$Q(\omega, \theta) = \frac{\Phi_u(\omega)}{|H(e^{j\omega}, \theta)|^2}$$

Therefore Q can be thought of as a weighting function that determines the bias distribution. Factors affecting this distribution will be discussed in Chapter 9 on experimental design.

A consistent estimator is one that generates estimates of the parameters that will converge to the actual value. Therefore, an estimate $\hat{\theta}$ of the MPUM θ_* is consistent if, in the long run (as $N \rightarrow \infty$) the difference between $\hat{\theta}$ and θ_* becomes negligible.

Variance Error. Estimated models are functions of random variables, and consequently, are always uncertain. Variance error is the measure of the distribution of the parameters about the mean. The quality of an estimator can be assessed by its mean-square error matrix [4]:

$$P = E\left\{\left[\hat{\theta}(y^N) - \theta_*\right]\left[\hat{\theta}(y^N) - \theta_*\right]^T\right\}$$

where θ_* is the "true value" of θ and is evaluated under the assumption that the PDF of y^N is $f_y(\theta_*; y^N)$. There is a lower limit to the values of P that can be obtained with any unbiased estimator. This limit is expressed in the Cramér-Rao inequality.

Let $\hat{\theta}(y^N)$ be an estimator of θ such that $E\{\hat{\theta}(y^N)\} = \theta_*$. Assume the PDF of y^N is $f_y(\theta_*; y^N)$, and suppose that y^N make take values in a subset of R^N whose boundary does not depend on θ . Then:

$$\text{Cov}(\hat{\theta}(y^N)) = E\left\{\left[\hat{\theta}(y^N) - \theta_*\right]\left[\hat{\theta}(y^N) - \theta_*\right]^T\right\} \geq M^{-1}$$

where

$$\begin{aligned} M &= E\left\{\left[\frac{d}{d\theta} \log f_y(\theta; y^N)\right] \left[\frac{d}{d\theta} \log f_y(\theta; y^N)\right]^T\right\}_{\theta=\theta_*} \\ &= E\left\{\frac{d^2}{d\theta^2} \log f_y(\theta; y^N)\right\}_{\theta=\theta_*} \end{aligned}$$

This bound applies for any N and all parameter estimation methods.

The Hessian $\frac{d^2}{d\theta^2} \log f_y(\theta; y^N)$ is a $d \times d$ matrix, the expected value of the Hessian matrix, the matrix M , is called the Fisher Information Matrix. If we let $\psi(t, \theta) = \frac{d}{d\theta} \hat{y}(t|\theta) = -\frac{d}{d\theta} \varepsilon(t|\theta)$, then for any unbiased estimator where $E\{\hat{\theta}(y^N)\} = \theta_*$, the Fisher Information Matrix becomes:

$$M = \frac{1}{K_0} \sum_{t=1}^N E\{\psi(t, \theta_*) \psi^T(t, \theta_*)\}$$

For gaussian innovations sequences, K_0 equals the variance. Therefore, in this case, we have:

$$\text{Cov}(\hat{\theta}(y^N)) \geq \kappa_0 \left[\sum_{t=1}^N E\{\psi(t, \theta_*) \psi^T(t, \theta_*)\} \right]^{-1}$$

Recall that we have assumed an unbiased estimator. Therefore, for the Cramér-Rao inequality to hold, we must have sufficient data and valid noise and model assumptions. In practice, parameter estimates resulting from linear gaussian models can exceed the Cramér-Rao lower bound by a factor of 5 to 10. This discrepancy is usually caused by the band limited nature of the noise power spectrum in contrast to the assumption of a flat spectrum.

Now we consider the variance of the transfer function as opposed to the variance in the parameters. Let Φ_u be the input spectrum, Φ_v the disturbance (noise) spectrum, and Φ_{uc} the cross-spectrum between the input and the innovations. Then we have for the covariance of the transfer function:

$$\text{Cov} \begin{bmatrix} \hat{G}(e^{j\omega}, \hat{\theta}_N) \\ \hat{H}(e^{j\omega}, \hat{\theta}_N) \end{bmatrix} \approx \frac{n}{N} \Phi_v(\omega) \begin{bmatrix} \Phi_u(\omega) & \Phi_{uc}(\omega) \\ \Phi_{uc}(-\omega) & \lambda_0 \end{bmatrix}^{-1}$$

An unbiased estimate is said to be efficient if its covariance equals the Cramér-Rao lower bound [5].

4.1.2. Validity Measures

As a first check of the result, use the estimate of the bias and the variance of the parameters to compute the confidence interval. If the confidence interval contains zero, we should consider whether this parameter should be removed.

If we assume that we have an efficient estimator and that the distribution of the parameters is gaussian, we can write expected value of the cost function as a conditional expected value involving the inverse of the covariance matrix. If we approximate this cost function by the first two terms of a Taylor series expansion we can write:

$$V(\theta) = V(\hat{\theta}) + \frac{1}{2} \partial \theta^T M \partial \theta$$

Therefore, we can define a confidence ellipsoid as the change $\partial \theta$ required to increase the cost function by a set amount. If we select $\Delta V(\theta) = \frac{1}{2}$, the confidence ellipsoid becomes $\partial \theta^T M \partial \theta = 1$. Since this ellipsoid often has many dimensions and cannot be expressed graphically, we must infer the shape from the length and direction of the principal axes. These are given by the eigenvalues and eigenvectors of M .

The insensitivity, I_i , gives the change in the parameter, $\partial \theta_i$, required to move from the minimum to the confidence ellipsoid and is given by $(M_{ii})^{-1}$. This value should be lower than the Cramér-Rao bound by a factor of two or more. An excessive sensitivity means that the response is insensitive to the parameter.

One method of separating the effects of scaling from the real problem of correlated parameters is to scale the information matrix by the diagonal matrix of the insensitivity so that all parameters have unit insensitivity. Let $SM = TMT$ with $T = \text{diag}(I_1, I_2, \dots, I_n)$. If two parameters are correlated they will share a high off-diagonal term in the scaled information matrix SM . If SM fails to show high correlations, we compare eigenvalues. Any problems will probably arise in the two largest parameters in the eigenvector that corresponds to the minimum eigenvalue.

4.2. Properties of the Identification Method

4.2.1. Least Squares Estimator

For the deterministic model $y = X\theta + \varepsilon$, $\hat{\theta}$ is an unbiased estimator

$$\begin{aligned} E\{\theta\} &= E\{(X'X)^{-1}X'y\} \\ &= E\{(X'X)^{-1}X'(X\theta + \varepsilon)\} \\ &= E\{(X'X)^{-1}X'XX'\theta + (X'X)^{-1}X'\varepsilon\} \\ &= \theta \end{aligned}$$

If measurement errors are present, the error terms are random variables that are a function of both the measurement noise and the choice of $\hat{\theta}$. The estimation error will be correlated with X , and the estimate will not be unbiased. Consequently, if the model includes measurement noise, the estimates will be biased. Consider the situation where the process model is $y_0 = X_0\theta + \varepsilon$, the output measurement error is $y = y_0 + V$ and the input measurements themselves have errors $X = X_0 + U$:

$$E\{\theta\} = -E\left\{\left[(X_0 + U)'(X_0 + U)\right]^{-1} [X_0 + U]'U\right\}\theta$$

We see that the bias is due solely to the correlation between modeling error and input measurement error.

4.2.2. Weighted Least Squares Estimator

The weighted least squares estimator is an unbiased estimate if the weight is equal to the inverse of the noise covariance. For the stochastic least squares estimator to be consistent, the following must hold:

1. The measurement noise must be nonsingular. In this case, the input is said to be persistently exciting.
2. Either the noise must be a sequence of zero mean, independent random variables (white noise), or the input sequence must be independent of the zero mean noise sequence. These conditions will insure that $E\{X(t)v(t)\} = 0$.

If the variance of the parameters is known, then we can expand the weighted least squares estimator to the minimum variance estimator.

4.2.3. General Prediction Error Method (PEM)

A quasi-stationary infinite data set is called "informative" if it allows us to distinguish between different models in a set. Given an informative data set and a uniformly stable, linear model structure, the PEM estimate will converge to the best possible approximation of the system that is available in the model set. If the actual system is within the model set, $\theta_* \in D_M$, then the distribution of errors will be asymptotically normally distributed, with zero mean and covariance P_0 . For a finite set of data points:

$$\sqrt{N}(\hat{\theta}_N - \theta_*) \in N(0, \hat{P}_N)$$

with

$$\hat{P}_N = \lambda_N \left[\frac{1}{N} \sum_{t=1}^N \psi(t, \hat{\theta}_N) \psi^T(t, \hat{\theta}_N) \right]^{-1}$$

where

$$\lambda_N = \frac{1}{N} \varepsilon^2(t, \hat{\theta}_N)$$

and

$$\psi(t, \hat{\theta}_N) = -\frac{d}{d\theta} \varepsilon(t, \hat{\theta}_N) = \frac{d}{d\theta} \hat{y}(t | \hat{\theta}_N)$$

4.2.4. Instrumental Variable (IV) Method

The quality of the instrumental variable (IV) method depends on the choice of instrumental variables. For the IV method to be successful, the instruments must be correlated with the regression variables, but uncorrelated with the noise.

Given an informative data set and a uniformly stable, linear model structure, the IV estimate will converge to the best possible approximation of the system that is available in the model set. If the actual system is within the model set, $\theta_* \in D_M$, then the distribution of parameter errors will be asymptotically normally distributed, with zero mean and covariance P_0 given by:

$$P_0 = \lambda_0 \left[E \{ \zeta(t, \theta_0) \psi^T(t, \theta_0) \} \right]^{-1} \left[E \{ \zeta(t, \theta_0) \psi^T(t, \theta_0) \} \right] \left[E \{ \zeta(t, \theta_0) \psi^T(t, \theta_0) \} \right]^{-T}$$

where $\theta_N = \theta_0$ results in $\varepsilon(t, \theta_0) = e_0(t)$ a sequence of zero mean independent random variables with covariance λ_0 .

4.2.5. Asymptotic Properties of Maximum Likelihood (ML) Estimators

Consider an estimate generated by a ML estimator. Assume that the observed random variables are independent and identically distributed so that:

$$f_z(\theta ; Z^N) = \prod_{i=1}^N f_z(\theta ; z_i)$$

and that the actual distribution of $\{z_i\}$ is defined by $f_z(\theta_* ; Z^N)$. Then $\hat{\theta}_{ML}(Z^N)$ tends to θ_* with probability 1 as $N \rightarrow \infty$, and $\sqrt{N}[\hat{\theta}_{ML}(Z^N) - \theta_*]$ converges to a zero mean normal distribution and covariance matrix given by the Cramér-Rao lower bound (M^{-1}). Therefore the $\hat{\theta}_{ML}(Z^N)$ is asymptotically unbiased and consistent.

5. MEASURES OF GLOBAL VALIDITY

While the local validity measures concentrate on internal measures of the model validity, the global measures are more focused on the ability of the model to represent the system. Again, there are two types of global validity measures. The first measure is with respect to the general information content in the data. Does the model extract the maximum amount of information from the data? Table 8.5.1 lists the information based validity measures we consider.

Table 8.5.1. Information Based Validation Methods.

TEST	COMMENTS
Akaike's information theoretic criterion	Requires the likelihood function
Final prediction error	Scales the loss function
Entropy	Requires the probability distributions

The second type of measure, data accuracy methods, attempts to measure the validity by computing the accuracy of the model output. The average and absolute model errors give a single measure of the accuracy of the model over the experimental region. Other tests measure the distribution and dispersion of the output errors, or the variance attributed to particular variables. Table 8.5.2 outlines some model validation methods.

Table 8.5.2. Data Accuracy Model Validation Methods.

TEST	COMMENTS
Maximum absolute error	Absolute value of the largest residual.
Average absolute error	Average value of the magnitude of the residuals.
Whiteness test	Residual analysis to insure that as much information as possible has been extracted from the data.
Lack-of-Fit test	Tests the order of the model with respect to the data.
Squared coefficient of determination	Ratio of the sum of the squares of the metamodel and the total sum of squares. Measures the proportion of the total variability in the response explained by the model. Does not measure the uniformity of the fit. Does not account for areas where there is no data.
Analysis of variance	Tests for the impact of additional variables.
Hypothesis testing	Tests the significance of the model parameters.

5.1. Information Based Model Accuracy Methods

5.1.1. Akaike's Information Theoretic Criterion (AIC)

As stated in Chapter 7, Akaike's information theoretic criterion (AIC) is:

$$AIC = -2 \max \left\{ \log \left(f_m(\hat{\theta}_N; N, Z^N) \right) \right\} + 2 \dim \theta$$

can be computed for different model structures where the criteria for the performance uses a log-likelihood error [6].

5.1.2. Akaike's Final Prediction Error (FPE)

The final prediction error (FPE) also provides a solution to the problem of model validity for auto-regressive models when least squares estimates are used [7]. For single-output systems the FPE is the one-step ahead prediction error and is defined as:

$$FPE = \frac{1 + \frac{n}{N}}{1 - \frac{n}{N}} V$$

where n is the number of parameters to be estimated, N is the length of the data record, and V is the loss function. The loss function depends on the Criterion of Fit (Chapter 6) and ranges from quadratic or robust norms to the maximum log likelihood function (LLF). For multi-output systems, the loss function is defined as the determinant of the estimated covariance matrix of the innovations.

5.1.3. Expected Relative Mutual Information.

Consider arbitrary random variables with an arbitrary joint density $p(c,d)$, and the normal densities $n(c,d)$, $n(c)$, and $n(d)$ defined by the first and second moments of the true distributions of c and d . The expected relative mutual information between the true density $p(c,d)$ and the approximating normal densities is [8]:

$$\int p(c,d) \left\{ \ln \frac{p(c,d)}{p(c)p(d)} - \ln \frac{n(c,d)}{n(c)n(d)} \right\} dcdd$$

5.2. Data Accuracy Model Validation Methods

5.2.1. Maximum Absolute Error

The maximum absolute error (MAE) is the absolute value of the largest residual. Using this criterion generates uniform but not necessarily good fits.

5.2.2. Average Absolute Error

The average absolute error (AAE) measures the average deviation from the actual data. This error measure will give an indication of the mean error in the estimate.

5.2.3. Average Absolute Relative Error

The formula for the average absolute relative error (AARE) is:

$$\text{AARE} = \frac{\sum_{i=1}^N |e_i / y_i|}{N}$$

While the AAE is independent of the magnitude of the data, this test scales the error by the magnitude of the output. This adjustment gives a better indication of the magnitude of the error relative to the data. A moderate error for a small data point will have a significant contribution to the AARE whereas only the raw magnitude is considered by the AAE.

5.2.4. Squared Coefficient of Determination

The squared coefficient of determination is:

$$R^2 = \frac{y' X (X' X)^{-1} X' y}{y' y}$$

It is a ratio of the sum of the squared of the metamodel and the total sum of squares. It measures the proportion of the total variability in the response explained by the model. The squared coefficient of determination does not measure the uniformity of the fit, nor can it account for areas where there is no data.

5.2.5. Residual Analysis

The assumption is that the errors are random, uncorrelated, and are normally distributed. These assumptions can be evaluated by the mean, standard deviation, power spectrum, and autocorrelation of the residuals.

Figure 8.5.1 is the plot of a random, uncorrelated, zero mean, and normally distributed sequence. The residuals should resemble this plot. Figure 8.5.2 is the plot of the power spectral density while Figure 8.5.3 is the plot of the autocorrelation of the sequence. The constant power spectrum and lack of correlation is evident.

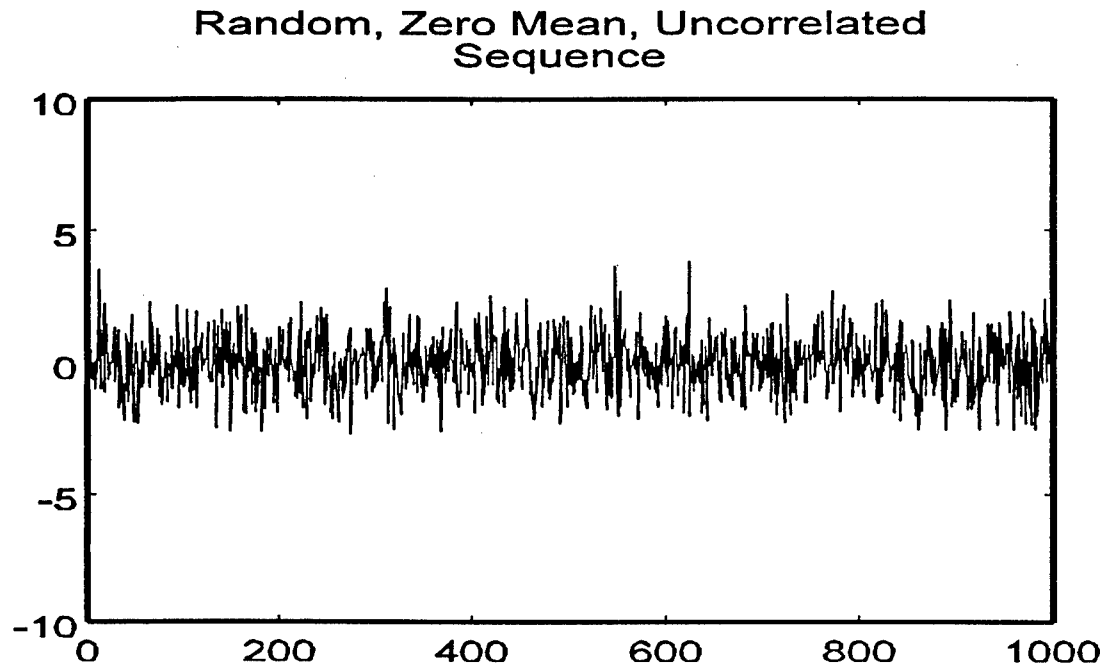


Figure 8.5.1. White Noise Sequence.

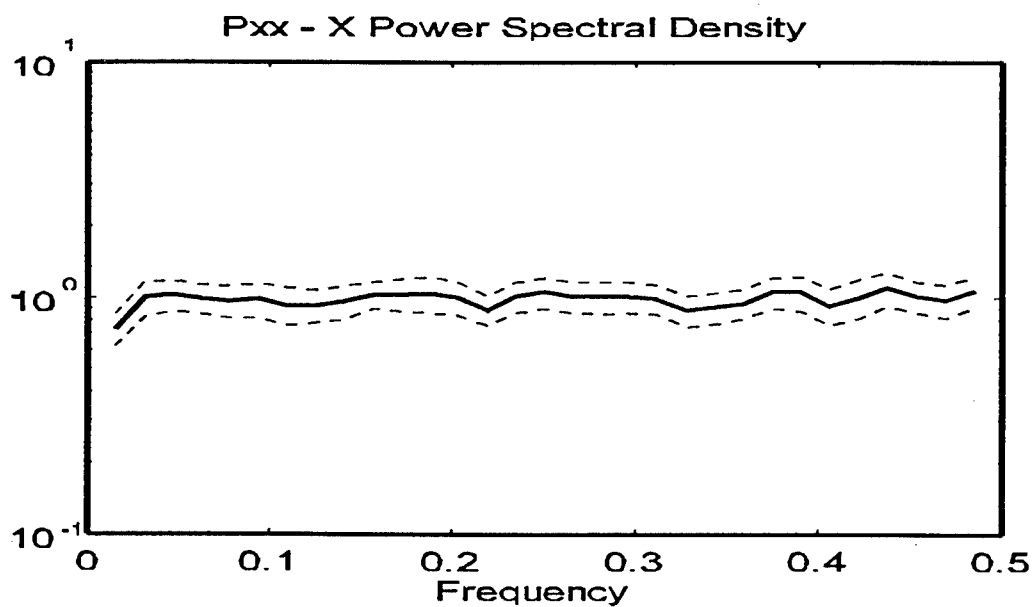


Figure 8.5.2. White Noise Power Spectral Density.

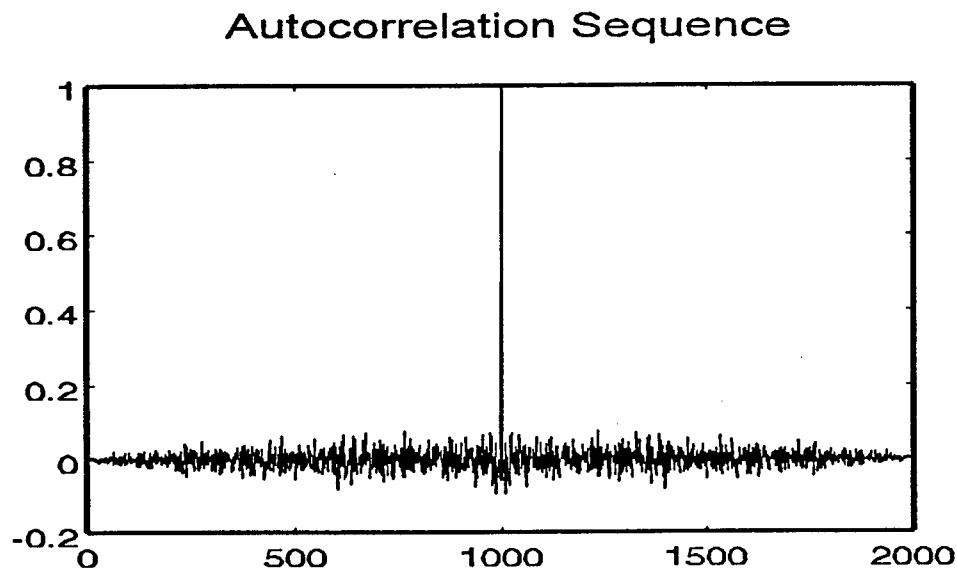


Figure 8.5.3. White Noise Autocorrelation Sequence.

The goodness of fit test is a hypothesis test where H_0 is the hypothesis that the model adequately fits the data. For residual analysis, we will determine if the residual sequence is normally distributed. There are several distribution dependent goodness of fit tests that can be used to determine if the residual sequence has a normal distribution. We present three tests. The first test is a whiteness test, the second a chi-Squared test of goodness of fit, the third is the Kolmogoroff-Smirnoff goodness of fit test.

5.2.5.1. Whiteness Test

The typical whiteness test determines the covariance of the residuals. For N data points, we compute $R_\varepsilon^N(\tau)$ for $\tau = 0, 1, 2, \dots, N-1$:

$$R_\varepsilon^N(\tau) = \frac{1}{N} \sum_{t=1}^{N-\tau} \varepsilon(t) \varepsilon(t+\tau)$$

If $\{\varepsilon(t)\}$ is a white-noise sequence, then

$$\zeta_{N,M} = \frac{N}{R(0)^2} \sum_{\tau=1}^M R_\varepsilon^N(\tau)^2$$

will be asymptotically $\chi_{M,\alpha}^2$ distributed [4]. Independence between signals can be tested by checking $\zeta_{N,M} < \chi_{M,\alpha}^2$.

5.2.5.2. Independence Between Residuals and Past Inputs

If the residuals are correlated with the past inputs then there is more in the output that originates from the input that is explained by the current model [4]. In addition to the

covariance of the residuals defined above $(R_e^N(\tau))$, define the covariance of the input $(R_u^N(\tau))$ in the same manner. Also define the crosscorrelation:

$$R_{eu}^N(\tau) = \frac{1}{N} \sum_{t=\tau}^N \varepsilon(t)u(t-\tau)$$

If $P = \sum_{k=-\infty}^{\infty} R_e^N(k)R_u^N(k)$, then the output residuals and inputs are independent if:

$$|R_{eu}^N(\tau)| \leq \sqrt{\frac{P}{N}} N_\alpha$$

where N_α is the α (significance) level of the normal distribution.

Correlation between the input and residual for negative τ is an indication of output feedback in the input, not a deficient model structure.

5.2.5.3. Chi-Squared Goodness of Fit

Assume that we have a sample from an unknown distribution F as well as a sample from a known distribution F_0 . The goodness of fit test assesses the null hypothesis of $H_0 : F(x) = F_0(x)$ against the alternate hypothesis $H_A : F(x) \neq F_0(x)$ [9]. The test statistic is:

$$\chi^2 = \sum_{i=1}^k \frac{(n_i - np_i)^2}{np_i} = \frac{1}{n} \sum_{i=1}^k \frac{n_i^2}{p_i} - n$$

where

- k = the number of classes each of sample of size n ,
- n_i = the observed frequency (number) of the class i ,
- np_i = the expected frequency (number) under H_0 , p_i is the probability of class i ,
- $v = k - 1 - a$, the degrees of freedom, where a is the number of parameters estimated from the sample.

We reject the H_0 hypothesis if $\chi^2 > \chi_{v,\alpha}^2$ where α is the significance of the Hypothesis test.

5.2.5.4. Kolmogoroff-Smirnoff (K-S) Goodness of Fit Test

While the Chi Squared test is better for detecting irregularities in the distribution, the K-S test is more sensitive to departures in the shape of the distribution [10]. This test is actually distribution free.

The null hypothesis, that the sample originated with a known distribution function, $F_0(x)$ is tested against the alternate hypothesis that the sample did not come from the

distribution. In this procedure, we sort the data by magnitude and the range of values is divided into classes. Form the observed absolute frequency distribution O and the frequency distribution, E , expected under the null hypothesis. Compute the cumulative frequencies in each of these classes, F_O and F_E . The test ratio is

$$D = \frac{\max|F_O - F_E|}{n}$$

Reject the hypothesis, at the significance level chosen, if D is greater than the bound from Table 8.5.3.

Table 8.5.3. Critical Values for the Test Statistic for the K-S Test.

BOUND ON D	SIGNIFICANCE LEVEL α
$1.073/\sqrt{n}$	0.20
$1.138/\sqrt{n}$	0.15
$1.224/\sqrt{n}$	0.10
$1.358/\sqrt{n}$	0.05
$1.628/\sqrt{n}$	0.01
$1.949/\sqrt{n}$	0.001

For $n > 30$, a comparison to the normal distribution (e.g., a whiteness test) can be made at:

$$0.805/\sqrt{n} (\alpha = 0.10), 0.866/\sqrt{n} (\alpha = 0.05), 1.031/\sqrt{n} (\alpha = 0.01)$$

5.2.6. Lack-of-Fit Test

This test, from [11], requires multiple input levers and repeated observations for each input. Consequently, it is only appropriate for multiple runs of stochastic systems. Assume that there are m levels.

The null hypothesis is that the model adequately fits the data, the alternate hypothesis is that the model does not fit the data. We will partition the residual sum of squares into $SS_E = SS_{PE} + SS_{LOF}$ where SS_{PE} is the pure experimental error, and SS_{LOF} is the sum of squares attributable to the lack of fit.

For each input level, compute the sum of the square error (difference between the individual output values and the mean) and then sum the square error over all input levels leading to:

$$SS_{PE} = \sum_{i=1}^m \sum_{j=1}^{n_i} (y_{ij} - \bar{y}_j)^2$$

There are $n - m$ degrees of freedom associated with the pure error sum of squares. The sum of squares for lack of fit is:

$$SS_{LOF} = SS_E - SS_{PE}$$

with $m - 2$ degrees of freedom. The test statistic for lack of fit is then:

$$F_0 = \frac{SS_{LOF} / (m - 2)}{SS_{PE} / (n - m)} = \frac{MS_{LOF}}{MS_{PE}}$$

and we reject the hypothesis if $F_0 > F_{\alpha, m-2, n-m}$.

5.2.7. Goodness of Fit Test

There are a number of distribution free tests that can be used to test the goodness of fit: the rank dispersion test of Siegel and Tukey; the U-test of Wilcoxon, Mann, and Whitney; the H-test of Kruskal and Wallis [10].

The sharpest homogeneity test is by Kolmogoroff-Smirnoff. It covers differences in the shape of the distribution, dispersion, skewness, and differences in the distribution function.

The greatest ordinate difference between the two empirical cumulative distribution functions serves as the test statistic. Data is sorted by magnitude and the range of values is divided into classes. The cumulative frequencies in each of these classes, F_1 and F_2 , are divided by the corresponding sample sizes n_1 & n_2 in the class. Then the differences are computed at regular intervals. The maximum of these values furnishes the test statistic D :

$$D = \max \left| \left(\frac{F_1}{n_1} - \frac{F_2}{n_2} \right) \right|$$

The critical value can be approximated for $n_1 + n_2 > 35$ by:

$$D_\alpha = K_\alpha \sqrt{\frac{n_1 + n_2}{n_1 n_2}}$$

where K represents a constant depending on the level of significance from Table 8.5.4:

α	0.20	0.15	0.10	0.05	0.01	0.001
K_α	1.07	1.14	1.22	1.36	1.63	1.95

Table 8.5.4. Values for K in the K-S Test.

Again the null hypothesis is rejected if the test statistic exceeds the critical value.

5.2.8. Analysis of Variance

The analysis of variance (ANOVA) is an extremely powerful tool that can be used to test the precision of the experiment, the significance of fewer coefficients, and adjusting the response for the effect of uncontrolled variables [12,13]. ANOVA is a statistical technique for analyzing measurements depending on several effects operating simultaneously and to decide which effects are important and to estimate the effects. The name is derived from a partitioning of the total variability into its component parts. The general ANOVA is integral to classical experimental design and the analysis depends on the experimental design and variance model used.

We, however, are concerned primarily with the adequacy of reduced order models to explain data generated by a higher order model. Therefore, we can consider the general linear regression significance test presented below to find the reduction in the total sum of squares that occurs from the reduced-order model.

5.2.9. Hypothesis Testing Regression Coefficients

Multiple linear regression model parameters hypothesis testing is based on the fact that the design matrix and output vector are not normalized by the σ^2 as in Chi-squared fitting.. The following hypothesis testing requires the assumption that the errors are $NID(0, s^2)$. This assumption makes the outputs $NID(b_0 + \sum_{i=1}^k b_i x_{ij}, s^2)$.

In testing for the significance of the regression, test:

$$H_0: b_1 = b_2 = \dots = b_k = 0$$

$$H_{1: b_i} \neq 0 \quad \text{for at least one } i$$

Rejection of H_0 implies that at least one variable in the model contributes significantly to the fit. The total sum of squares is partitioned into regression and error sums of squares:

$$SS_{yy} = SS_R + SS_E$$

If $H_0: b_i = 0$ is true, then $SS_R / \sigma^2 \approx \chi_k^2$ where the number of degrees of freedom is equal to the number of regressor variables. Also, $SS_E / \sigma^2 \approx \chi_{n-k-1}^2$, and SS_E and SS_R are independent.

Therefore the test procedure for $H_0: b_i = 0$ is to compute:

$$F_0 = \frac{SS_R / k}{SS_E / (n - k - 1)} = \frac{MS_R}{MS_E}$$

and to reject $H_0: b_i = 0$ if $F_0 > F_{\alpha, k, n-k-1}$.

Testing individual regression coefficients is useful in determining the value of the significance of the variates. The model may be more effective with the addition or deletion of one or more variables. Adding a variable to the regression model always causes the sum of the squares for the regression to increase and the error sum of squares to decrease. The decision is whether the increase in the regression sum of squares is sufficient to warrant the additional variables. Adding an unimportant variable can actually increase the mean square error.

Since the least squares estimator b is a random variable, a linear combination for the observations, the distribution $b \approx N[b, \sigma^2(X'X)^{-1}]$ (assuming that the design and output data has not been normalized by the standard deviation of the measurement). Therefore, each regression coefficient has the property $b_i \approx N[b_i, \sigma^2 C_{ii}]$, where C_{ii} is the $(i+1)^{st}$ diagonal element (starting from b_0) of $(X'X)^{-1}$.

The hypothesis for testing the significance of an individual coefficient then is:

$$H_0: b_i = 0$$

$$H_1: b_i \neq 0$$

The appropriate test statistic for independent b_i is $t_0 = \frac{\hat{\beta}}{\sqrt{MS_E C_{ii}}}$, and $H_0: b_i = 0$ is rejected if $|t_0| > t_{\alpha/2, n-k-1}$.

The contribution given that other parameters are already in the model can be determined by partitioning the b vector into:

$$b = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

where b_1 is $(r \times 1)$ and b_2 is $(p-r) \times 1$ resulting in the model $y = X_1 b_1 + X_2 b_2 + \epsilon$. To find the contribution of the terms b_1 , we fit the model assuming that $H_0: b_i = 0$ to be true. The reduced model is $y = X_2 b_2 + \epsilon$, the least squares estimator of b_2 is $b_2 = (X_2' X_2)^{-1} X_2' y$, and $SS_R(b_2) = b_2' X_2' y$.

The regression sum of squares due to b_1 , adjusted for the presence of b_2 already in the model is $SS_R(b_1|b_2) = SS_R(b) - SS_R(b_2)$. Since $SS_R(b_1|b_2)$ is independent of MS_E , the null hypothesis $H_0: b_i = 0$ can be tested by the statistic:

$$F_0 = \frac{SS_R(b_1|b_2)/r}{SS_E/(n-k-1)} = \frac{SS_R(b_1|b_2)/r}{MS_E}$$

If $F_0 > F_{\alpha, r, n-p-1}$, reject $H_0: b_i = 0$ and conclude that at least one of the parameters in b_1 is not zero.

6. BLOCK STRUCTURED, NORM-BOUNDED UNCERTAINTY

The most common error model used for identification assumes that all of the errors enter the system as additive noise [14]. This model does not adequately account for the perturbations in the model parameters. In recent years, advances in robust control techniques have been built on a block structured, norm-bounded uncertainty that enters the model in a linear fractional manner.

Inclusion of this research into a metamodeling procedure is a major topic for future research.

7. REFERENCES

1. Press, et al, *Numerical Recipes*, Cambridge University Press, 1986.
2. E. Lefeber, and J.W. Polderman, "On the Possible Divergence of the Projection Algorithm," *IEEE Transactions on Automatic Control*, Vol. 40, No. 3, March 1995.
3. Franklin, Powell, Workman, *Digital Control of Dynamic Systems*, 2nd Edition, Addison-Wesley Publishing Co., Reading, MA, 1990.
4. L. Ljung, *System Identification: Theory for the User*, Prentice-Hall, New Jersey, 1987.
5. J. P. Norton, *An Introduction to Identification*, Academic Press, San Diego, CA, 1986.
6. G. C. Goodwin, M. Gevers, and Brett Ninness, "Quantifying the Error in Estimated Transfer Functions with Application to Model Order Selection," *IEEE Trans. on Automat. Contr.*, vol 37, no. 7, pp 913-928, July 1992.
7. H. Akaike, "Canonical Correlation Analysis of Time Series and the Use of an Information Criterion," *System Identification: Advances and Case Studies*, R.K. Mehra and D.G. Lainiotis, eds. Academic Press, New York, pp 27-96., 1976.
8. W. E Larimore, "System Identification and Filtering of Nonlinear Controller Markov Processes by Canonical Variate Analysis," Final Report for AF Office of Scientific Research, October 27, 1989.
9. H. Cramer, *Mathematical Methods of Statistics*, Princeton University Press, 1963.
10. L. Sachs, *Applied Statistics*, Springer-Verlag, New York, 1984.
11. D. C. Montgomery, *Design and Analysis of Experiments*, John Wiley & Sons, NY, 1991.
12. Engineering Design Handbook U.S. Army Material Command Pamphlet AMCP 706-110, December 1969.
13. H. Scheffe, *The Analysis of Variance*, Wiley Publications in Statistic, New York, 1959.
14. Smith, R.S. and Doyle, J., "Model Validation: A Connection Between Robust Control and Identification," *IEEE Transactions on Automatic Control*, Vol. 37, No. 7, July 1992.

CHAPTER 9

EXPERIMENTAL DESIGN

1. CHAPTER OUTLINE

1. Chapter Outline	9-1
2. Introduction.....	9-2
2.1. General Discussion.....	9-2
2.2. Identifiability and Observability	9-3
2.3. Minimal Realizations	9-4
2.4. Input-Output Requirements.....	9-5
2.5. Closed-Loop Experiments.....	9-6
2.6. Discrete Event Systems	9-7
2.7. Summary.....	9-8
3. Statistical Experimental Design.....	9-9
3.1. Guidelines for Statistical Experimental Design.....	9-9
3.2. Transformations	9-10
3.3. Regression Diagnostics.....	9-11
3.3.1. Detecting Influential Observations and Outliers.....	9-12
3.3.1.1. Single Row Effects on the Coefficients.....	9-12
3.3.1.2. Single Row Effects on the Fitted Values.....	9-13
3.3.1.3. Single Row Effects on the Covariance Structure of the Coefficients.....	9-13
3.3.1.4. Single Row Effects on the Variance of the Estimate	9-13
3.3.1.5. Geometric Approaches.....	9-13
3.3.1.6. Multiple Row Effects on the Fitted Values	9-14
3.3.1.7. Multiple Row Effects on the Covariance Structure of the Coefficients	9-14
3.3.2. Detecting and Assessing Collinearity.....	9-14
3.3.2.1. Objectives of the Analysis	9-15
3.3.2.2. Historical Procedures to Detect Collinearity:.....	9-15
3.3.2.3. Background	9-16
3.3.2.4. Impact of Condition Number on the Regression Result.....	9-17
3.3.2.5. Procedure	9-18
3.4. Corrective Measures	9-20
4. Experimental Design for Identification	9-21
4.1. General	9-21
4.2. Guidelines for Experimental Design for Identification	9-22
4.3. Design Criteria	9-27
5. Input-Output Variable Selection for Metamodeling.....	9-29
5.1. General	9-29
5.2. Analytical Metamodels	9-29
5.3. Simulation Metamodels	9-29
5.4. Existence of a True Input-Output Relationship	9-29
6. References.....	9-32

2. INTRODUCTION

This chapter is concerned with issues that are pertinent to Step 10. **"Select an Experimental Design."** The chapter first discusses the general issue of "What properties of the behavior allow the system to be properly represented by a difference (or differential) equation of a particular type?" Then we discuss aspects of experimental design from both a statistical and engineering perspective.

Classical "experimental design" for statistical methods is the process of designing the experiment so that appropriate data can be analyzed by statistical methods that require identically independently distributed (IID) random variables.

Experimental design for identification is concerned with model set and structure selection, identification criterion, identifiability, and validation of the model.

2.1. General Discussion

The design of an experiment includes which variables to measure and when to measure them and which variables to manipulate and how to manipulate them. Experimental design structures the change to the input variables so that we may observe and identify the reasons for changes in the output response. How this is accomplished depends on your point of view.

An analyst (or statistician) will spend significant time deciding how to draw a sample from the general population so that the data will conform to certain assumptions and allow valid statistical inference. Control engineers take a different tact. They are usually trying to identify a model for a piece of equipment and will concentrate on insuring a persistently exciting input signal so that all of the system modes will be excited. We must combine both elements of experimental design.

By combining elements of both disciplines, we overcome weaknesses in both areas. From a statistical perspective, anyone who has analyzed data knows that it is possible to correlate two variables when there is no logical or mathematical reason to believe that such a relationship exists [1]. Looking at Figure 9.2.1, we see why this can be the case. The input is $u(t)$ and the output is $z(t_i)$. However, the relationship between the two is defined by the combination of four time varying processes: input, disturbance, system, and measurement. By concentrating on the identification of these internal processes, we analyze the system at a level of detail well within the input and output.

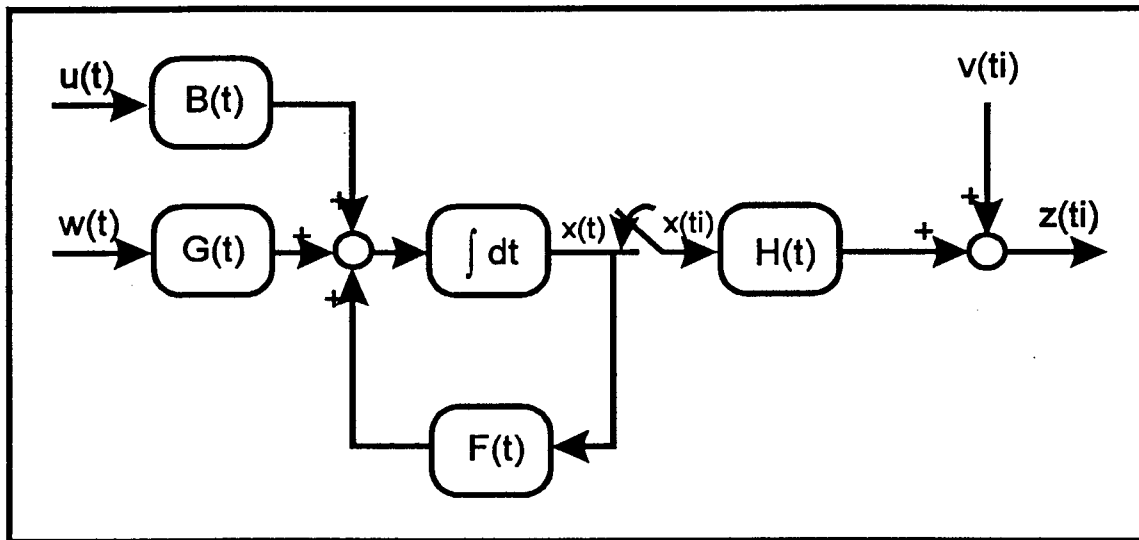


Figure 9.2.1. General Time Varying Linear System Description.

While the statistical techniques are generally straightforward, well defined, and mathematically sound, system identification, as practiced by control engineers, has yet to arrive at a general unifying procedure. Instead, a myriad of techniques, each best for a particular situation, have defined the art of identification. By applying a structured experimental design, we can develop a set of procedures usable for many different situations.

2.2. Identifiability and Observability

Consider the general scalar model structure:

$$A(t)y(t) = \frac{B(q)}{F(q)}u(t) + \frac{C(q)}{D(q)}e(t)$$

This model is locally and globally identifiable at θ_* if and only if [6]:

1. There is no common factor to all of $z^{n_a}A_*(z)$, $z^{n_b}B_*(z)$, $z^{n_c}C_*(z)$.
2. There is no common factor to $z^{n_b}B_*(z)$ and $z^{n_r}F_*(z)$.
3. There is no common factor to $z^{n_c}C_*(z)$ and $z^{n_d}D_*(z)$.
4. If $n_s \geq 1$, then there is no common factor to $z^{n_r}F_*(z)$ and $z^{n_d}D_*(z)$.

In the absence of process noise and *a priori* information the continuous Ricatti equation is [2]:

$$\dot{P}(t) = F(t)P(t) + P(t)F^T(t) - P(t)H^T(t)R(t)^{-1}H(t)P(t)$$

using the matrix identity $\dot{P}^{-1} = -P^{-1}\dot{P}P^{-1}$, the solution to this linear equation is:

using the matrix identity $\dot{P}^{-1} = -P^{-1} \dot{P} P^{-1}$, the solution to this linear equation is:

$$P^{-1}(t) = \int_0^t \Phi(\tau, t) H^T(\tau) R^{-1}(\tau) \Phi(\tau, t) d\tau$$

If the integral is positive definite for some $t > 0$ then $P^{-1}(t) > 0$ and the system can decrease the estimation error variance. For discrete-time systems the condition for uniform complete observability is:

$$\alpha I \leq \sum_{i=k-N}^k \Phi(i, k) H^T R^{-1} \Phi(i, k) \leq \alpha_2 I$$

2.3. Minimal Realizations

Depending on the model structure, the normal equations (least squares) provide a unique solution [14]. To obtain a unique parameter set, we must select a canonical form having a minimum number of parameters. A parameter set θ with one and only one value of $\hat{\theta}$ that minimizes the error criterion is said to be identifiable. Two parameters that minimize the error criterion are said to be equivalent.

Given a state space system (A, B, C, D) , we can obtain a minimal realization having the same transfer function by first removing any uncontrollable modes and then removing unobservable modes [3]. We describe the staircase algorithm which is numerically stable.

Obtain a SVD of B : $B = U_1 \Sigma_1 V_1^H$, and obtain a row compression¹:

$$U_1^H B = \begin{bmatrix} Z_1 \\ 0 \end{bmatrix}$$

in which Z_1 has full row rank. Let

$$U_1^H A U_1 = \begin{bmatrix} Y_1 & X_1 \\ B_1 & A_1 \end{bmatrix}$$

in which X_1 and Y_1 have the same number of rows as Z_1 , and A_1 and Y_1 are square.

Now obtain a SVD of B_1 : $B_1 = U_2 \Sigma_2 V_2^H$ and form the state transformation $U_1 \begin{bmatrix} I & 0 \\ 0 & U_2 \end{bmatrix}$

which transforms the pair (B, A) into the pair:

$$\left(\begin{bmatrix} Z_1 \\ 0 \end{bmatrix}, \begin{bmatrix} Y_1 & X_1 U_2 \\ Z_2 & X_2 \\ 0 & B_2 & A_2 \end{bmatrix} \right)$$

¹The superscript H denotes transpose of the complex conjugate.

This process is repeated until it terminates with $B_k = 0$ (defined by comparing singular values to a preset threshold or machine precision) or B_k having full row rank. Therefore, we can define the transformation:

$$T_i = \begin{bmatrix} I & 0 \\ 0 & U_i \end{bmatrix}$$

with $T = T_1 T_2 \dots T_k$ to provide the following:

$$\begin{bmatrix} I & 0 \\ 0 & T^{-1} \end{bmatrix} \begin{bmatrix} D & -C \\ B & sI - A \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & T \end{bmatrix} = \begin{bmatrix} D & -C_c & \vdots & * \\ B_c & sI - A_c & \vdots & * \\ \dots & \vdots & \dots & \vdots \\ 0 & 0 & \vdots & sI - A_k \end{bmatrix}$$

so that (A_c, B_c, C_c, D) is a controllable realization with the same transfer function as (A, B, C, D) .

The algorithm can be continued to remove any unobservable modes. This is accomplished by using the same transformation T to transform the pair $\begin{bmatrix} C_c^T & A_c^T \end{bmatrix}$ into:

$$\begin{bmatrix} [C_{co} & 0] \\ [A_{co} & 0] \\ [* & A_i] \end{bmatrix}$$

which also transforms matrix pencil as shown below:

$$\begin{bmatrix} D & -C_c \\ B_c & sI - A_c \end{bmatrix} \Rightarrow \begin{bmatrix} \begin{bmatrix} D & -C_{co} \\ B_{co} & sI - A_{co} \end{bmatrix} & \vdots & 0 \\ \dots & \dots & 0 \\ 0 & 0 & \vdots & sI - A_i \end{bmatrix}$$

2.4. Input-Output Requirements

Having defined the system and its behaviors as in Chapter 3, we can address this key issue of the inverse modeling problem. We can define "What properties of the behavior allow the system to be properly represented by a difference (or differential) equation of a particular type?"

It can be shown that for a system to be represented by means of a difference (or differential) equation, it has to be **complete** (it cannot have initialization or termination conditions at $t = \pm \infty$) with a **finite memory span** so that observation of a trajectory on a finite time interval allows conclusions about past behavior independent of what will happen in the future [4]. With proper experimental design, the input-output spaces will have a definite structure [5]. The presence of this structure can be used as the basis for a test to determine appropriate experimental design.

Data used for the generation of a reduced order metamodel has to be "informative." A quasi-stationary infinite data set is called "informative" if it allows us to distinguish between different models in a set. A quasi-stationary infinite data set is informative if the spectrum matrix for $z(t) = [u(t)y(t)]^T$ is strictly positive definite for all ω [6]. An open loop experiment is informative if the input signal is persistently exciting.

A persistently exciting input is a sequence $\{u(k)\}$ that fluctuates sufficiently to avoid the possibility that only linear combination of the elements of θ will show up in the error criterion [14]. The structure of the input and output sets of the metamodel also has a structure that will be discussed below.

2.5. Closed-Loop Experiments

Information generated by closed-loop experiments could easily be defective. Consider the first order ARX model shown in Figure 9.2.2 with a constant, linear regulator:

$$\begin{aligned}y(t) + ay(t-1) &= bu(t-1) + e(t) \\ u(t) &= fy(t)\end{aligned}$$

Incorporating the feedback, we get the closed-loop model:

$$y(t) + (a - bf)y(t-1) = e(t)$$

Therefore, all models

$$\hat{a} = a + \gamma f$$

$$\hat{b} = b + \gamma$$

with γ an arbitrary constant will give the same input-output description. Consequently, even with a persistently exciting input, there is no way to distinguish between these models even if we know the regulator parameter f .

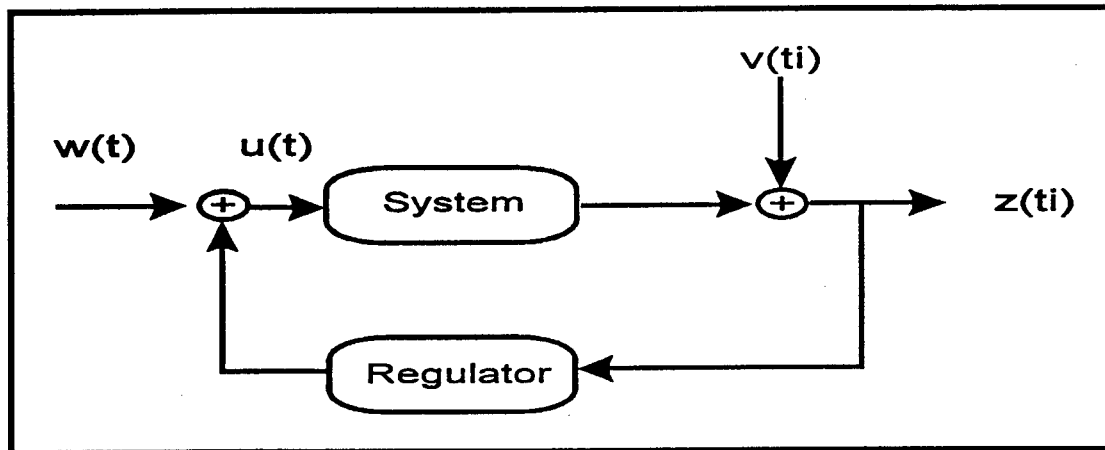


Figure 9.2.2. Closed-Loop Experimental Setup.

If the regulator is nonlinear, time-varying, noisy, or complex, however, the experiment should be informative enough. Consider the following input:

$$u(t) = F_i(q)y(t) + G_i(q)\omega(t), \quad i = 1, 2, \dots, k$$

where each of the regulators is stable. Then the experiment is informative if and only if:

$$\det \begin{bmatrix} \sum_{i=1}^k G_i(e^{j\omega}) G_i^T(e^{j\omega}) + F_i(e^{j\omega}) F_i^T(e^{j\omega}) & \sum_{i=1}^k F_i(e^{j\omega}) \\ \sum_{i=1}^k F_i^T(e^{j\omega}) & k \cdot I \end{bmatrix} > 0$$

Therefore, we conclude that we can make a closed-loop experiment informative by switching between linear regulators or by adding an extra signal that passes through filters without zeros on the unit circle.

2.6. Discrete Event Systems

The question of the experimental design is complicated by the fact that the simulations (the entity that is the basis for the metamodel) are usually realizations of discrete event systems (DES).

The framework outlined in Chapter 3 is consistent with the formalized discrete-event systems in theoretical computer science. The behavior is similar to the formal language; a state-space system is like an automation; latent variables are replaced by production rules; interconnections are communications. The most significant difference is the lack of behavioral models (equations) in the theory of a DES. Also, completeness is usually violated in a DES by initiation and termination rules for event strings. The question arises: "When can a DES be described by a difference equation?" This chapter will directly address this question.

2.7. Summary

In summary, assuming that the underlying system modeled by the simulation is well behaved (Markovian, complete with respect to the modeled behavior), the following is required to metamodel combat simulations:

1. The data must include the behavior we are trying to model.
2. The latent variables that define the behavior must be observable.
3. The input must be persistently exciting so that the effects of the latent variables are observed.
4. For a stochastic system, the ensemble of trajectories must span the space.
5. Any single trajectory must span both the input and output space and be sufficiently long so that the state transition probabilities also span the allowable probability space and the distribution of these probabilities are the same as the underlying system.

Given that issues associated with behavioral properties and the DES nature of the simulation are correctly addressed, a combination of statistical and identification experimental design will be used. In every case, a proper design of experiments is required. If a linear regression is used, techniques of regression diagnostics are appropriate.

3. STATISTICAL EXPERIMENTAL DESIGN

Classical "experimental design" are the methods used to structure an experiment, test, or series of tests. The purpose of the structure is to make purposeful changes in the input variables so that we may observe and identify the reasons for changes in the response [7,8].

Since statistical methods are used, classical "experimental design" is the process of designing the experiment so that appropriate data can be analyzed by statistical methods that require identically independently distributed (IID) random variables [9,10,11]. The three basic principles of experimental design are replication, randomization, and blocking [8].

Replication permits the estimation of experimental error while reducing it. Randomization permits an unbiased estimation of the effects by the elimination of known and unknown systematic errors in particular trends. This principle brings about the independence of the test results. Block division increases precision within the blocks. Within a block, randomization applies. With blocking, nuisance factors are eliminated by analysis of covariance when the factors are known and nonmeasurable perturbing factors are defined by formation of the block groups.

Usually it is not possible to absolutely insure that the variables are IID. Experimental designs have been devised to improve the precision by which comparisons are made. These designs consist of single factor, randomized blocks, latin squares (Graeco-Latin square), factorial, nested (hierarchical), and response surface methods. When one of these designs is used for modeling, two techniques -- analysis of variance analysis (ANOVA) and residual analysis (or a variation thereof) -- are used to determine the adequacy of a model fit.

3.1. Guidelines for Statistical Experimental Design

The experimental design process can be summarized by the following guidelines:

Clearly recognize and state the problem.

Choose the Factors and Levels. This requires knowledge of the process. For example in the output error model of the TERSM system presented in Volume 2, there is no data that characterizes the emitter field, and it would not be possible to include a factor for different emitter fields. These metamodels pertain to the given emitter field only.

Selection of the Response Variables. The response variables are a function of both the process and of the use of the metamodel. For simulation metamodels, both the input factors and response variables are defined by the coupling requirements. Although, it is possible to break a large problem into a series of smaller ones so that

the probability of a causal relationship is higher. For analytical metamodels the response factors should be closely coupled to the input factors.

Choose an Experimental Design. Select a sample size, run order, and blocking or randomization procedure. Reference 7 presents randomized, block, factorial, and nested design procedures. Keep the design and analysis as simple as possible.

Perform the Experiment. Here it is important to monitor the results to insure that the experimental design is being followed. Remember that experiments are iterative. Validation may be required to confirm the experiment.

Analyze the Data. Good engineering and process knowledge should be combined with statistical techniques to insure that the process results in a metamodel that adequately represents the higher order model. Recognize the difference between practical and statistical significance.

3.2. Transformations

Often the relationships that we are trying to identify are nonlinear. Nonlinear transformation of variables is a commonly used practice in regression to stabilize the error variance or to normalize the error distribution. Least squares is predicated on linear independence on the input variables and a linear additive relationship between the inputs and the outputs. Therefore, another goal of transformations, and the reason we introduce this subject, is to find transformations that produce the best-fitting additive model. Also, knowledge of the transformations aids in the interpretation and understanding of the relationship between response and predictors.

If $\phi_i(X_i)$ is a function of the input, the fraction of variance not explained by a regression of $\theta(Y)$ on $\sum_{i=1}^p \phi_i(X_i)$ is:

$$\epsilon^2(\theta, \phi_1, \dots, \phi_p) = \frac{E\left\{\theta(Y) - \sum_{i=1}^p \phi_i(X_i)\right\}^2}{E\{\theta^2(Y)\}}$$

Optimal transformations exist, and is one that minimizes this error [12]. An algorithm to determine this optimal transformation involves alternating conditional expectations (ACE). There are SISO, MISO, and MIMO versions of the algorithm. For MISO systems, the algorithm is:

Set $\theta(Y) = Y/|Y|$ and $\phi_1(X_1), \dots, \phi_p(X_p) = 0$

Iterate until $\varepsilon^2(\theta, \phi_1, \dots, \phi_p)$ fails to decrease

Iterate until $\varepsilon^2(\theta, \phi_1, \dots, \phi_p)$ fails to decrease

For $k = 1$ to p

$$\phi_{k,1}(X_k) = E \left\{ \theta(Y) - \sum_{i \neq k} \phi_i(X_i) \mid K_k \right\}$$

replace $\phi_k(X_k)$ with $\phi_{k,1}(X_k)$

end of for loop

end of inner iteration loop

$$\theta_1(Y) = E \left\{ \sum_{i=1}^p \phi_i(X_i) \mid Y \right\} / \left\| E \left\{ \sum_{i=1}^p \phi_i(X_i) \mid Y \right\} \right\|$$

replace $\theta(Y)$ with $\theta_1(Y)$

end outer iteration loop

The algorithm may be extended to MIMO systems minimizing $E \left\{ \sum_{i=1}^n \theta_i(Y_i) - \sum_{i=1}^p \phi_i(X_i) \right\}$

by adding an inner loop over the response variables.

3.3. Regression Diagnostics

In this approach, the entire data set is generated (to the highest order desired) and then statistical procedures are used to explore characteristics of the data for a regression application [13]. The purpose of this technique is to detect influential observations and outliers by identifying subsets of data that have a disproportionate influence on the model. It should be noted that these outlying data points may contain valuable information.

Procedures for this type of diagnostics include: row deletion (observation) effect on coefficients, fitted values, residuals, and on the covariance structure of the coefficients; sensitivity to small perturbations by differentiation of regression outputs with respect to model parameters; geometric approaches such as Wilk's Δ statistic and generalized distance metrics (Mahalanobis distance). Since one outlier can mask the effect of another, multiple row effects tests must also be included.

A second set of procedures is designed to detect and assess collinearity which is the primary source of ill conditioning among regression variables. These procedures consist of an analysis of the singular values judged to have a high condition index. This procedure identifies variables that are associated with high variance-decomposition proportions for two or more estimated regression coefficient variances.

3.3.1. Detecting Influential Observations and Outliers

An influential observation is one which, either individually or together with several other observations, has a demonstrably larger impact on calculated values of the estimates. The distinction between influential observation and outlier is difficult because the extreme data points may contain valuable information that improves estimation efficiency by its presence. Influential data points should be removed only if it can be shown that they are in error. The usual validation methods focus on the residuals or the error in the predicted values. While they can identify problems, they cannot provide information as to what the model would look like if the input data set was modified in some manner, nor can they provide the effect of individual data points on the overall model.

Before fitting the metamodel, we should explore the characteristics of the data that will be used. In addition to the univariate distribution, the following techniques are designed to identify subsets of data that have a disproportionate influence on models

3.3.1.1. Single Row Effects on the Coefficients

The effect of a single observation (row) on the coefficients is measured by estimating the change in the coefficient if it were deleted. This effect is:

$$\Delta\theta_i = \frac{\varepsilon_i \sqrt{N}}{(N-1)s(i)}$$

where ε_i is the residual of the row, and the estimate of the variance (stochastically independent of the numerator) is:

$$s(i) = \frac{1}{N-p-1} \sum_{k \neq i} [y_k - x_k(\theta(i))]^2$$

with p = the number of columns of the inputs including a column of ones.

The scaled (with respect to the variance of the component) measure of change in a single component is:

$$\Delta\theta_{ij} = \frac{c_{ji}}{\sqrt{\sum_{k=1}^N (c_{jk})^2}} \frac{\varepsilon_i}{s(i)(1-h_i)}$$

with c_{ji} the components of $C = (X^T X)^{-1} X^T$ which is proportional to the variance of θ_j and the (diagonal elements of the projection) $h_i = x_i (X^T X)^{-1} x_i^T$. Large value of $\Delta\theta_{ij}$ indicate that this observation is influential in determining the j_{th} component.

3.3.1.2. Single Row Effects on the Fitted Values

The change in the fit is defined as:

$$\Delta_{FIT} = \hat{y}_i - \hat{y}_i(i) = \frac{h_i \varepsilon_i}{1 - h_i}$$

which can be scaled by the standard deviation of the fit to give:

$$\Delta_{FITS} = \left[\frac{h_i}{1 - h_i} \right]^{1/2} \frac{\varepsilon_i}{s(i)\sqrt{1 - h_i}}$$

3.3.1.3. Single Row Effects on the Covariance Structure of the Coefficients

Comparing the covariance matrix when all data is used to the covariance matrix with a deleted row by the ratio of the determinants scaled by variance leads to:

$$COVRATIO = \frac{1}{\left[\frac{N - p - 1}{N - p} + \frac{\varepsilon_i^{*2}}{N - p} \right]^p (1 - h_i)}$$

where $\varepsilon_i^* = \frac{\varepsilon_i}{s(i)\sqrt{1 - h_i}}$ is the studentized residual. This measure should be near unity and we should investigate points with $[COVRATIO - 1] \geq 3p/N$.

3.3.1.4. Single Row Effects on the Variance of the Estimate

A ratio for assessing the effect of a single row on the variance of the estimate is:

$$VAR = \frac{s(i)^2}{s^2(1 - h_i)}$$

which provides similar results to a ratio of the determinants of the differentiated coefficients.

3.3.1.5. Geometric Approaches

A geometric look at the effects of the projection matrix and the residuals is offered by forming a matrix $Z = [X \ y]$ and considering Wilk's Δ statistic:

$$\Lambda(\tilde{z}_i) = \frac{\det(\tilde{Z}^T \tilde{Z} - (N - 1) \tilde{Z}^T(i) \tilde{Z}(i) - \tilde{z}_i^T \tilde{z}_i)}{\det(\tilde{Z}^T \tilde{Z})}$$

where \tilde{Z} is the centered Z matrix. This statistic can be shown equal to:

$$\Lambda(\tilde{z}_i) = \left(\frac{N}{N-1} \right) (1 - h_i) \left(1 + \frac{\varepsilon_i^{*2}}{N-p-1} \right)^{-1}$$

and can be related to the F statistic by:

$$\left(\frac{N-p-1}{p} \right) \left(\frac{1 - \Lambda(\tilde{z}_i)}{\Lambda(\tilde{z}_i)} \right) \approx F_{p, N-p-1}$$

Another approach is to evaluate the Mahalanobis distance between one row and the mean of the rest:

$$M(\tilde{z}_i) = (N-2)(\tilde{z}_i - \tilde{z}(i)) \left(\tilde{\tilde{Z}}(i)^T \tilde{\tilde{Z}}(i) \right)^{-1} (\tilde{z}_i - \tilde{z}(i))^T$$

where $\tilde{\tilde{Z}}(i)$ is $\tilde{Z}(i)$ centered by $\tilde{Z}(i)$.

3.3.1.6. Multiple Row Effects on the Fitted Values

With D_m the set of rows to be deleted, the change in the fit for the data points remaining after deletion is:

$$\Delta_{\text{FITS}}(D_m) = \varepsilon_{D_m}^T X_{D_m} \left[X^T(D_m) X(D_m) \right]^{-1} X_{D_m}^T \varepsilon_{D_m}$$

When D_m is used as a subscript, it denotes a matrix or vector with rows whose indices are in D_m . A stepwise approach begins with selecting the two largest Δ_{FITS} to form $D_2^{(1)}$ and computing:

$$\Delta_{\Delta\text{FIT}} = \left| x_k \left[\theta - \theta(D_2^{(1)}) \right] \right|$$

If the two largest values do not have their indices in $\Delta_{\Delta\text{FIT}}$, a set $D_2^{(2)}$ is formed consisting of the indices for the two largest. This procedure is iterated until a set D_2 is formed coinciding with the two largest values of $\Delta_{\Delta\text{FIT}}$.

3.3.1.7. Multiple Row Effects on the Covariance Structure of the Coefficients

The single row COVRATIO can be extended to multiple rows by:

$$\text{COVRATIO} = \frac{\det s^2(D_m) \left[X^T(D_m) X(D_m) \right]^{-1}}{\det s^2(X^T X)^{-1}}$$

3.3.2. Detecting and Assessing Collinearity

Here, we look for sources of ill conditioning among regression variables and assess the extent that the least-squares estimate $\theta = (X^T X)^{-1} X^T y$ is potentially harmed by collinear relations. An example of these procedures is presented in Volume II, Chapter 1.

The harm from collinearity comes from the fact that a collinear relation can result in a situation where the observed influence of the explanatory variables is overcome by the model's random error term, reducing the signal to noise. Collinearity is ill conditioning, a data problem. Variates are collinear if the data vectors lie in a subspace of dimension less than the number of variates. This is equivalent to saying that there is a high correlation between the variates. Collinearity will be defined in terms of the conditioning of the data matrix X .

3.3.2.1. Objectives of the Analysis

1. How many near dependencies plague a given data set.
2. Which variates have coefficient estimates adversely affected by the presence of the dependencies.
3. Whether estimates of interest are included among those with inflated confidence intervals and, therefore, corrective action is warranted.
4. Whether prediction intervals based on the estimated model are greatly inflated by the presence of ill conditioned data.
5. Whether specific coefficient estimates of interest are relatively isolated from the ill effects of collinearity and, therefore, trustworthy in spite of ill conditioned data.

3.3.2.2. Historical Procedures to Detect Collinearity:

First we present five standard procedures to detect collinearity with a discussion of the test.

1. Variables have low t -statistics, and various regression results are sensitive to the deletion of a row or column of X .

DISCUSSION: None of these conditions is necessary or sufficient for the existence of collinearity.

2. Examining the correlation matrix or inverse correlation matrix of the explanatory variables. The diagonal matrix of R^{-1} , r_{jj} are called the variance inflation factors (VIF).

DISCUSSION: A high correlation coefficient or VIF can point to a collinearity problem, the absence of a high correlation cannot be viewed as no problem. Three or more variates can be collinear, while no two of the variates taken alone can be highly correlated.

3. Farrar and Glauber's technique: This technique is a measure of collinearity based on the assumption that X is a sample of size n from a p -variate Gaussian distribution. Under the assumption that X has orthogonal columns, the transformation $\det(R)$ is approximately χ^2 distributed and provides a measure of collinearity.

DISCUSSION: Like \mathbf{R} , $\det(\mathbf{R})$ cannot diagnose several near dependencies. Also, \mathbf{X} is fixed, not stochastic, and collinearity is not a statistical phenomenon.

4. Bunch-map analysis: A graphical investigation of possible relationships.

DISCUSSION: Addresses the first diagnostic problem - the location of the dependencies - but does not make an attempt to determine the degree to which the results are impacted by their dependence. Its extension to more than two variates is time consuming.

5. Eigenvalues and eigenvectors of the correlation matrix: Collinearity is indicated by the presence of a small eigenvalue of $\mathbf{X}^T \mathbf{X}$.

DISCUSSION: There is not a definition of small. A standard for small has become the other eigenvalues which shows the relevance of the condition number of the matrix.

3.3.2.3. Background

The spectral norm is defined as $\|\mathbf{A}\| \equiv \sup_{\|z\|=1} \|\mathbf{A}z\|$. In addition to its usual properties as a norm, it obeys the following property (which follows directly from its definition): $\|\mathbf{A}z\| \leq \|\mathbf{A}\| \cdot \|z\|$.

A matrix \mathbf{A} is ill conditioned if the product of its spectral norm with that of \mathbf{A}^{-1} is large. This measure is the condition number of \mathbf{A} . The larger the condition number, the more ill condition the matrix.

Consider the system of equations $\mathbf{A}z = \mathbf{c}$, where \mathbf{A} is square, nonsingular with solution $z = \mathbf{A}^{-1}\mathbf{c}$. How much does the solution vector z change (δz) if there were small perturbations in the elements of \mathbf{A} or \mathbf{c} ?

If \mathbf{A} is fixed but \mathbf{c} changes by $\delta \mathbf{c}$, then we have $\delta z = \mathbf{A}^{-1}\delta \mathbf{c}$. Therefore, we can write $\|\delta z\| \leq \|\mathbf{A}^{-1}\| \cdot \|\delta \mathbf{c}\|$. But since $\|\mathbf{c}\| \leq \|\mathbf{A}\| \cdot \|z\|$ we have:

$$\frac{\|\delta z\|}{\|z\|} \leq \|\mathbf{A}\| \cdot \|\mathbf{A}^{-1}\| \cdot \frac{\|\delta \mathbf{c}\|}{\|\mathbf{c}\|}$$

Therefore, $\|\mathbf{A}\| \cdot \|\mathbf{A}^{-1}\|$ provides a bound for the relative change in the solution vector from a relative change in \mathbf{c} . For perturbations in the elements of the matrix \mathbf{A} :

$$\frac{\|\delta z\|}{\|z + \delta z\|} \leq \|\mathbf{A}\| \cdot \|\mathbf{A}^{-1}\| \cdot \frac{\|\delta \mathbf{A}\|}{\|\mathbf{A}\|}$$

Consequently, because of its usefulness, $\|A\| \cdot \|A^{-1}\|$ is defined as the condition number, $\kappa(A)$, of the nonsingular matrix A .

Any $n \times p$ matrix A may be decomposed as $A = UDV^T$ where $U^T U = V^T V = I$ and D is diagonal with nonnegative diagonal elements μ_k , $k = 1, \dots, p$ called singular values of A . Since $A^T A = VD^2 V^T$, the singular value decomposition (SVD) also provides information that encompasses the eigensystem $A^T A$. The orthogonal columns of V are the eigenvectors of $A^T A$ and the columns of U are the p eigenvectors of AA^T associated with the p non-zero eigenvalues.

If there are exact linear dependencies, the rank of A will be less than p . Since U and V are orthogonal (and of full rank) the $\text{rank}(A) = \text{rank}(D)$. Partitioning the SVD as

$$A = UDV^T = U \begin{bmatrix} D_{11} & 0 \\ 0 & 0 \end{bmatrix} V^T$$

with $V_1 \Rightarrow p \times r$, $U_1 \Rightarrow n \times r$, $V_2 = p \times (p - r)$, and $U_2 \Rightarrow n \times (p - r)$ results in the two equations:

$$\begin{aligned} AV_1 &= U_1 D_{11} \\ AV_2 &= 0 \end{aligned}$$

and V_2 provides an orthonormal basis for the null space of A .

Returning to the spectral norm of A , it is well known that $\|A\| = \mu_{\max}$, the maximum singular value of A . Also, if A is square $\|A^{-1}\| = 1/\mu_{\min}$. Therefore, the condition number may be computed as:

$$\kappa(A) = \frac{\mu_{\max}}{\mu_{\min}} \geq 1$$

3.3.2.4. Impact of Condition Number on the Regression Result

In general, the condition number gives an estimate of the significance of the data. If data are known to d significant digits, and the condition matrix of the regressor matrix is 10^r , then the solution is significant to $(d-r)$ digits.

In addition, the SVD can be used to determine the location and impact of any near dependencies.

The variance-covariance matrix of the least-squares estimator $\theta = (X^T X)^{-1} X^T y$ is $\sigma^2 (X^T X)^{-1}$ where σ^2 is the common variance of the components in the linear model. Using the SVD $X = UDV^T$, the variance-covariance matrix $V(\theta)$ is:

$$V(\theta) = \sigma^2 (X^T X)^{-1} = \sigma^2 V D^{-2} V^T$$

so that for the k^{th} component of θ ,

$$\text{var}(\theta_k) = \sigma^2 \sum_j \frac{v_{kj}^2}{\mu_j^2}$$

is decomposed into a sum of components each associated with one and only one of the p singular values. Consequently, components associated with small singular values (near dependencies) will be large relative to the other components.

To exploit this relationship to examine the effects on regression estimates, define the k_j^{th} variance-decomposition proportion (π) as the proportion of the variance of the k^{th} regression coefficient associated with the j^{th} component in the variance-covariance matrix:

$$\pi_{jk} \equiv \frac{\frac{v_{kj}^2}{\mu_j^2}}{\sum_{j=1}^p \frac{v_{kj}^2}{\mu_j^2}}$$

From the above discussion, we see that there are two components to the variance; Collinearity (represented by μ_j^2) and orthogonality (represented by v_{kj}^2). If two nearly collinear variates are mutually orthogonal then the variance of the coefficients would be unaffected.

Also, it should be noted that the conditioning of the data is a strong function of the parameterization (structure). Reparameterization could improve the conditioning, but it is a function of the singular values (dependencies) of the data.

3.3.2.5. Procedure

The test diagnostic is:

A singular value judged to have a high condition index, and which is associated with high variance-decomposition proportions for two or more estimated regression coefficient variances.

For the purposes of analyzing collinearity, it is always desirable to scale X to have equal (unit) column lengths. This scaling does not change the parameterization but just changes the units in which the X variates are measured.

If the data contain a constant term, X should contain uncentered data along with a column of ones. The use of centered data should be avoided since centering can mask the role of

the constant in any underlying near dependencies and produce misleading diagnostic results.

STEP 1. Scale the data matrix X to have unit column length.

STEP 2. Compute the SVD of X :

- a. Calculate the condition indices from $\eta_k = \frac{\mu_{\max}}{\mu_k}$, $k = 1, \dots, p$
- b. Calculate the Π matrix of variance-decomposition proportions from

$$\pi_{jk} \equiv \frac{\frac{v_{kj}^2}{\mu_j^2}}{\sum_{j=1}^p \frac{v_{kj}^2}{\mu_j^2}}$$

STEP 3. Determine the number and relative strengths of the near dependencies by the condition indices exceeding a given threshold.

- STEP 4.
- a. Examine the condition indices for the presence of competing dependencies; roughly equal condition indices.
 - b. Examine the condition indices for the presence of dominating dependencies; high condition indices (exceeding the threshold in STEP 3) coexisting with even larger indexes.

STEP 5. Determine the involvement of the near dependencies. For this analysis a threshold for $\pi = \pi^*$, must be established. $\pi^* = .5$ seems to work well. Three cases must be considered.

- a. Only one near dependency: There is a single near dependency if two or more variates have variance-decomposition proportions greater than the threshold. Only one high variance-decomposition proportion will not result in degradation.
- b. Competing dependencies: When two or more near dependencies are competing, then the high variance-decomposition proportions involved in the separate competing dependencies can be arbitrarily distributed among them. The number of coexisting dependencies or the variates in the competing dependencies is still recoverable, only the information on the separate involvement of specific variates in competing dependencies is lost. With competing dependencies the variance-decomposition proportions that exceed the threshold are aggregated over all competing dependencies.
- c. Dominating Dependencies: In this case, auxiliary regressions are warranted. We cannot rule out the involvement of a given variate (cannot

assume the noninvolvement) even if it is the only one with a high variance-decomposition proportion.

STEP 6. Once a number of near dependencies have been determined, auxiliary regressions should be run. Beginning with the strongest near dependency, pick as dependent variables the one(s) that is(are) strongly involved in an underlying near dependency to regress separately on the remaining variables.

STEP 7. Determine variates that remain unaffected by the presence of collinear relations.

3.4. Corrective Measures

For least squares estimators, there are a number of corrective measures available to improve model validity.

1. Introduction of new data.

2. Bayesian-type techniques.

a. **Pure Bayes.** This requires addition of subjective prior information on the parameters of the model and an exact statement of the prior distribution.

b. **Mixed Estimation.** Prior or auxiliary information is added directly to the data matrix. Beginning with the linear model: $y = X\beta + \varepsilon$, construct $r < p$ prior restrictions on the elements of β of the form $c = R\beta + \xi$. This results in the augmented matrix equation:

$$\begin{bmatrix} y \\ c \end{bmatrix} = \begin{bmatrix} X \\ R \end{bmatrix} \beta + \begin{bmatrix} \varepsilon \\ \xi \end{bmatrix}$$

where

$$V \begin{pmatrix} \varepsilon \\ \xi \end{pmatrix} = \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix}$$

If Σ_1 and Σ_2 are known, generalized least squares results in the unbiased mixed-estimation estimator:

$$b_{me} = (X^T \Sigma_1^{-1} X + R^T \Sigma_2^{-1} R)(X^T \Sigma_1^{-1} y + R^T \Sigma_2^{-1} c)$$

c. **Ridge Regression.** The ridge regression estimator with a single ridge parameter k is $b = (X^T X + kI)^{-1} X^T y$. This is equivalent to mixed estimation with $R=A$ (with $A^T A = I$) $\Sigma_1 = \sigma^2 I$, $\Sigma_2 = \lambda^2 I$, and $c = 0$ so that $k = \sigma^2 / \lambda^2$. In mixed estimation c is taken to be stochastic, whereas here c is taken as a set of constants, which results in a biased estimator.

4. EXPERIMENTAL DESIGN FOR IDENTIFICATION

4.1. General

All real world processes are nonlinear infinite dimensional dynamical systems. When we attempt to model specific behaviors of these processes, we develop an approximation to the process. Chapters 5, 6, 7, and 8 provided structures and techniques to arrive at and validate this approximation. All of these result in finite dimensional approximations.

In addressing this problem, we have two choices. We can attempt to directly identify a nonlinear system. In this case, we combine the input variables in a nonlinear fashion to arrive at the output. If we do not want to work with the nonlinear system, we can attempt to identify a series of linear systems that together generate the nonlinear behavior that we observe.

The objective of the metamodeling procedure is to represent the high fidelity model (simulation) as close as possible. If we assume, as in Chapter 7, that the high fidelity model can be represented as:

$$y(t) = G_o(q)U(t) + H_o(q)e_o(t)$$

with

$$T_o(q) = [G_o(q) \ H_o(q)]$$

we can measure the metamodel as $\hat{T}(q, \hat{\theta}_N) = [\hat{G}(q, \hat{\theta}_N) \ \hat{H}(q, \hat{\theta}_N)]$ and the difference between the high fidelity model and the metamodel as:

$$\tilde{T}(e^{j\omega}, \hat{\theta}_N) = \hat{T}(e^{j\omega}, \hat{\theta}_N) - T_o(e^{j\omega})$$

Introducing a frequency weighting for the identification problem the design criterion becomes:

$$J(\hat{\theta}_N) = J_P(\hat{\theta}_N) + J_B(\hat{\theta}_N)$$

$$J_P(\hat{\theta}_N) = \frac{1}{N} \int_{-\pi}^{\pi} \text{tr} [P(\omega, \hat{\theta}_N) C(\omega)] d\omega$$

$$J_B(\hat{\theta}_N) = \int_{-\pi}^{\pi} B(e^{j\omega}, \hat{\theta}_N) C(\omega) B^T(e^{-j\omega}, \hat{\theta}_N) d\omega$$

where

$$B(e^{j\omega}, \hat{\theta}_N) = \tilde{T}(e^{j\omega}, \hat{\theta}_N) - T_o(e^{j\omega})$$

$$P(\omega, \hat{\theta}_N) = T^T(e^{-j\omega}, \hat{\theta}_N) [N \cdot \text{Cov}\{\hat{\theta}_N\}] T(e^{j\omega}, \hat{\theta}_N)$$

and the objective is to minimize J.

4.2. Guidelines for Experimental Design for Identification

The first principle for the experimental design for identification is to use all available prior knowledge to reduce the uncertainty in the estimate [14]. Referring to Figure 9.2.2, we see the complexity of even a relatively simple system. Consequently, we want to partition the behaviors (systems) into components small enough so that we can measure the observable states. This partition is limited by the fact that we need to make sure that the behaviors we are trying to represent are complete.

The guidelines for this type of experimental design are grouped by the choices available to the analyst. Each of these decisions will have an influence on the quality of the resulting model.

The choices for the experimental design for identification can be grouped as follows:

A. Input-Output Data

1. Select dependent signals (outputs)

This includes what to measure and where to measure it.

2. Select driving signals to measure (inputs)

The inputs determine the operating point of the system and which modes of the system are excited during an experiment. Interesting parameters must have a clear effect on the output predictions. In addition to the inputs for the experiment, we must also identify input signals that cannot be changed but can be measured. They can either be included as inputs or disturbances for the identification.

3. Select sampling interval

Sampling leads to information loss. If T is the sampling interval then $\omega_s = 2\pi/T$ is the sampling interval and the Nyquist frequency is $\omega_N = \omega_s/2$. Information at a frequency twice the Nyquist frequency cannot be distinguished from information below that frequency. Consequently, high frequency contributions are aliased into the lower part of the spectrum and the information is lost.

Sampling at high frequencies compared to the natural frequency of the process is numerically sensitive. A suitable choice of sampling frequency is approximately ten times the bandwidth of the system. If this is not possible, the input signal should be filtered with a low pass (antialiasing) filter to remove frequency components above the Nyquist frequency.

4. Select input characteristics (spectra)

In addition to the actual input, the spectrum and the cross spectrum between the input and driving noise is important. Recall that the weighting function that determines experimental design was found to be:

$$Q(\omega, \theta) = \frac{\Phi_u(\omega)}{|H(e^{j\omega}, \theta)|^2}$$

Therefore the selection of the input signal spectrum $\Phi_u(\omega)$ will affect the bias in the estimates.

Identifiability is insured by a persistently exciting input without simple feedback mechanisms. A quasi-stationary input signal $\{u(t)\}$ with spectrum $\Phi_u(\omega)$ is persistently exciting of order n if, for all filters of the form:

$$M_N(q) = m_1 q^{-1} + \dots + m_N q^{-N}$$

the result:

$$|M_N(e^{j\omega})|^2 \Phi_u(\omega) = 0$$

implies that $M_N(e^{j\omega}) = 0$. Therefore, $\{u(t)\}$ is persistently exciting if $\Phi_u(\omega)$ is different from zero for at least N points on the interval $-\pi < \omega \leq \pi$. Also, a persistently exciting signal cannot be filtered to zero by an $(N-1)$ th-order moving average filter.

For least squares, an input is persistently exciting if the lower right $(n \times n)$ -matrix component of $X^T X$ (which depends only on the sequence $\{u(k)\}$) is nonsingular. A discrete signal is persistently exciting of order n if its discrete spectrum is nonzero at least n points over the range $0 < \omega T < \pi$. Also, one would like to design the experiment so that $X^T X$ is diagonal. Since off-diagonal terms in $X^T X$ are the sums of cross products, an orthogonal design will accomplish this. The 2^k factorial design is an orthogonal design [1]. Given the assumptions that go into the model the primary issues for least squares estimation is the inequality of variance and correlated measurements.

5. Choose the number of samples to collect.

B. Treatment of Data

Deficiencies in the data can come from high frequency disturbances (noise), occasional bursts and outliers, and drift and offset (low frequency disturbances). It is especially important to remove trends and drifts when a fixed noise model is used. These deficiencies can be handled in one of three ways. First, the data can

be pretreated. Second, the data can be filtered. Third, we can let the noise model take care of the disturbances. Here we discuss four methods to pretreat the data.

1. Let the input and output be deviations from equilibrium. Here we determine the input and output levels that correspond to operating point. Subtract these values from the data so that the data becomes the deviations from this equilibrium.
2. Subtract sample means. This is one method of determining the operating point.
3. Estimate offsets and drifts explicitly. This is a slight variant of the second approach by modeling the system with a constant that is estimated.
4. Difference the data. Differencing the data (both the input and the output) is equivalent to prefiltering with the filter $L(q) = 1 - q^{-1}$ or using a noise model with integration.

C. Model Set and Structure

1. Model type
2. Model class
3. Model order

If the model orders are overestimated global and local identifiability will be lost and the information matrix will be singular. Therefore, the condition number of the information matrix can be used as an indication of model order.

4. Model

- Predictor

- Transfer functions
- Noise models

The selection of the noise model set $H_1(q, \theta)$ that is to represent the observed noise characteristics will also affect the bias distribution.

If the data has a constant bias, use a noise model with integration (this is equivalent to differencing the data). The noise model can be extended by allowing higher order terms.

- Probabilistic

D. Identification Criterion

1. Prediction error methods

- Criterion of Fit (Norm)

The choice of the norm only acts as an independent scaling of the covariance matrix of the parameters. Selection of the optimal norm requires knowledge of the true covariance.

- Prefilter

The first step in the PEM was to filter the prediction sequence $\varepsilon(t, \theta)$ using a stable linear filter $L(q)$:

$$\varepsilon_F(t, \theta) = L(q)\varepsilon(t, \theta) \quad 1 \leq t \leq N$$

This filter will affect the observed noise distribution $H(e^{j\omega}, \theta)$ and the bias in the estimates. Depending on the frequency of the pass band, the filter can shape the performance of the identification in different frequency ranges.

If the information signal is band limited, and the noise has a broadband signal, then an antialiasing filtering will remove some of the noise, improve the signal to noise ratio of the input, and reduce bias. If the information is not bandlimited, then the antialiasing filter will also remove some of the information in the signal. When an antialiasing filter is used, include it in the model of the system.

- Prediction horizon

The prediction horizon will affect the observed noise distribution and the bias just as the prefilter. Usually, as k increases, the weighting function:

$$W_1(e^{j\omega}, \theta) = \frac{1}{H(e^{j\omega}, \theta)}$$

becomes more of a low pass filter.

- Numerical procedure

2. Correlation methods

- Correlation vectors
- Prefilter
- Shaping function
- Numerical procedure

3. Probabilistic models

Probabilistic models require a maximum likelihood approach, are computationally expensive, and can exhibit numerical difficulties (convergence

problems in estimating the measurement or process noise covariance matrix) when the assumptions do not match the data. The MMLE formulation from Chapter 6 provides the most robust approach. Details are in [15].

With this approach, selection of the initial process and measurement noise covariances must be added to the list of choices. The added burden of identifying the Kalman gain and measurement and process noise can reduce the identifiability of the model structure.

Use of process noise should be used with care because a Kalman filter is required to estimate the states and the gain of the filter can become high. This causes the state error to decay rapidly emphasizing low frequency behavior and giving high frequency modeling errors greater opportunity to influence parameter values.

E. Validation measure

1. Select the procedure and criterion by which the metamodel will be validated. These choices were discussed in Chapter 8.

4.3. Design Criteria

The experimental condition should resemble the situation for which the model is to be used. The bias of the estimator, $B(e^{j\omega}, \hat{\theta}_N)$, should be addressed first. This will insure an informative experiment. The bias in the transfer function and noise model estimates is entirely or partially determined by a certain weighting function, $Q(\omega, \hat{\theta}_N)$, which is the result of the choice of noise model, input spectrum, input-noise cross spectrum, prefilter, and prediction horizon.

The weighting function $Q(\omega, \hat{\theta}_N)$ will not be known *a priori* since it depends on the parameter estimates. It is, however, always available posteriori. Therefore, even if we cannot predict the effect of a particular selection, we can see the results of the selections, or a change in a selection, after the identification. Just as in classical experimental design, in off-line applications one should first look for deficiencies by plotting the data.

The signal-to-noise ratio of the input signal should be chosen proportional to the criterion weighting function. This can be accomplished by input design, noise model selection, or prefiltering.

Once we have an informative experiment and the bias is acceptable, design parameters can be tuned to minimize the covariance matrix $P(\omega, \hat{\theta}_N)$. A small variance in a certain component results if the predictor is sensitive to that component. Therefore, outputs and inputs should be selected so that the predicted output becomes sensitive to parameters that are important for the application. This is the issue of optimal input design discussed next.

Minimization of the parameter covariance matrix is equivalent to the maximization of the average information matrix per sample. Averaging the Fisher information matrix from Chapter 8, we have:

$$\bar{M} = \lim_{N \rightarrow \infty} \frac{1}{N} M_N = \frac{1}{K_0} \sum_{t=1}^N \bar{E} \{ \psi(t, \theta_0) \psi^T(t, \theta_0) \}$$

This can be written as:

$$\bar{M}(\Phi_u) = \int_{-\pi}^{\pi} M(\omega) \Phi_u(\omega) d\omega + \lambda_0 M_e$$

where λ_0 is the variance of the zero mean residual sequence, the primes are gradients of G and H , and:

$$\tilde{M} = \frac{(\lambda_0/K_0) \left(1/2\pi\right) G'_\theta(e^{j\omega}, \theta_0) \left[G'_\theta(e^{-j\omega}, \theta_0) \right]^T}{\Phi_v(\omega)}$$

$$M_e = \frac{\lambda_0}{2\pi K_0} \int_{-\pi}^{\pi} \frac{H_0'(e^{j\omega}, \theta_0) [H_0'(e^{-j\omega}, \theta_0)]^T}{\Phi_v(\omega)} d\omega$$

To achieve a large information matrix, input energy should be concentrated at frequencies where \tilde{M} is large. If a parameter is of special interest, vary it, check the amplitude of the transfer function (Bode plot) and place the input energy there.

The average information matrix depends on the actual or true system parameters. Since we use estimates of the true system parameters that are constrained by the selected model structure and representation, these selections will constrain the design.

This optimization of the average information matrix is usually carried out with a scalar measure of \bar{M} [16]. We will consider $\alpha(\bar{M}(t, \theta)) = \text{tr}[\bar{M}^{-1}(t, \theta) \cdot C]$ where C is derived from the frequency weighting criterion:

$$J_p(\hat{\theta}_N) = \frac{1}{N} \int_{-\pi}^{\pi} \text{tr}[P(\omega, \hat{\theta}_N) C(\omega)] d\omega$$

From Chapter 8, we have for the covariance of the transfer function:

$$\text{Cov} \begin{bmatrix} \hat{G}(e^{j\omega}, \hat{\theta}_N) \\ \hat{H}(e^{j\omega}, \hat{\theta}_N) \end{bmatrix} \approx \frac{n}{N} \Phi_v(\omega) \begin{bmatrix} \Phi_u(\omega) & \Phi_{uc}(\omega) \\ \Phi_{uc}(-\omega) & \lambda_0 \end{bmatrix}^{-1}$$

with Φ_u be the input spectrum, Φ_v the disturbance (noise) spectrum, and Φ_{uc} the cross-spectrum between the input and the innovations. Applying the covariance of the transfer function leads to the following minimization:

$$J_p(\omega, \{\Phi_u(\omega), \Phi_{uc}(\omega)\}) = \int_{-\pi}^{\pi} \Psi(\omega, \{\Phi_u(\omega), \Phi_{uc}(\omega)\}) d\omega$$

with

$$\Psi(\omega, \{\Phi_u(\omega), \Phi_{uc}(\omega)\}) = \frac{\lambda_0 C_{11}(\omega) - 2 \text{Re}[C_{12}(\omega) \Phi_{uc}] + C_{22}(\omega) \Phi_u}{\lambda_0 \Phi_u(\omega) - |\Phi_{uc}(\omega)|^2} \Phi_v(\omega)$$

If $\Phi_{uc}^{\text{opt}} \equiv 0$, then the optimal input is

$$\Phi_u^{\text{opt}}(\omega) \equiv \mu_1 \sqrt{C_{11}(\omega) \Phi_v(\omega)}$$

and

$$|H_{\text{opt}}(e^{j\omega})|^2 = \mu_2 \sqrt{\frac{\Phi_v(\omega)}{C_{11}(\omega)}}$$

where μ_1 is a constant adjusted so that $\int_{-\pi}^{\pi} \Phi_u^{\text{opt}}(\omega) d\omega = a$ and μ_2 is a constant such that is monic.

5. INPUT-OUTPUT VARIABLE SELECTION FOR METAMODELING

5.1. General

Recall that the definition of a dynamical system was defined as a family of trajectories without reference to input-output maps. In fact, the input and the output of a process (simulation component) could be the same set of variables with different values. As a simple example, consider a simulation module that updates the position of an aircraft as some function of the aircraft state. The input will be the aircraft state (positions, velocities, accelerations, fuel, etc.) the output will be the aircraft state.

5.2. Analytical Metamodels

The input and output of an analytical metamodel is constrained by the simulation and defined by the analyst. The analyst must select variables that exist within the simulation. Which of the variables selected by the analyst is a function of the analytical requirements that caused the generation of the metamodel in the first place.

In some cases the I/O may be undefined. Here the analyst is looking for general relationships in the relative importance of variables and a general search over the entire space may be required.

Once selected, the purpose of the metamodel defines the required domain and range of the metamodel. One factor that affects this selection is the level of the simulation.

5.3. Simulation Metamodels

A metamodel used to support hierarchical simulation by **tearing** is constrained by both the simulation itself and the additional simulation components that will be coupled to the metamodel. The external simulations also define the domain and range of the metamodel.

A metamodel used to support the **synthesis** of hierarchical simulations has additional freedom. In this case, the number of simulation components that will be included in a single metamodel define the constraints. A single metamodel may be made of a single simulation component or of coupled components. In each case the processes included in the metamodel will vary (although) the input and output variables of the coupled metamodel may not.

5.4. Existence of a True Input-Output Relationship

Assume that we have observed the input and output of a system and computed a set of linear differential and/or algebraic equations from this data. Have we identified the system? Do these equations establish a true input-output relationship suggested by this identification? Answers to this question are provided by two sequences of subspaces, one in the input space u and the other in the output space y [5].

Consider a system on linear ordinary differential and algebraic equations with constant coefficients: $A(\sigma)\xi + B(\sigma)u + C(\sigma)y = 0$ where σ denotes differentiation (or the shift operator for discrete time systems), and ξ contains all of the latent variables not present in the input and output spaces. $A(s)$, $B(s)$, and $C(s)$ are polynomial matrices.

We say that y processes u if the linear space of trajectories $\{y | (y, 0) \in \mathcal{B}\}$ is finite dimensional. Therefore y processes u if u determines y up to a finite number of constants. Also, u is free if for every trajectory u there exists a trajectory y such that $(y, u) \in \mathcal{B}$.

Recall that if the dynamical system with latent variables $\Sigma_f = (Z, R^q, R^d, B_f)$ is linear time invariant and complete, then the manifest system which it represents $\Sigma = (Z, R^q, B)$ is also linear time invariant and complete. Consequently, for a linear time invariant and complete system, any behavior given by $A(\sigma)\xi + B(\sigma)u + C(\sigma)y = 0$ can also be represented by

$$B = \left\{ \begin{bmatrix} y \\ u \end{bmatrix} \begin{bmatrix} R_1(\sigma) & R_2(\sigma) \end{bmatrix} \begin{bmatrix} y \\ u \end{bmatrix} = 0 \right\}.$$

The behavior of such a set of equations stems from an input-output system if both conditions of the following proposition hold.

Proposition VII: Let a behavior B be given by

$$B = \left\{ \begin{bmatrix} y \\ u \end{bmatrix} \begin{bmatrix} R_1(\sigma) & R_2(\sigma) \end{bmatrix} \begin{bmatrix} y \\ u \end{bmatrix} = 0 \right\}$$

where $\begin{bmatrix} R_1(\sigma) & R_2(\sigma) \end{bmatrix}$ is a polynomial matrix of full row rank. The following statements hold:

1. y processes u if and only if $R_1(s)$ has full column rank,
2. u is free if and only if $R_1(s)$ has full row rank.

Therefore $R_1(s)$ must be invertible and the transfer matrix of the system is defined by $T(s) = -R_1(s)R_2(s)^{-1}$.

Theorem VIII: Let the behavior with external variables y and u be given by the pencil form:

$$\begin{aligned} \sigma Gz &= Fz \\ w &= Hz \end{aligned}$$

²Note: The frequency response determines the transfer function and the transfer function determines the controllable part of the system. In general, however, neither the frequency response nor the transfer function determines the behavior completely.

with the vector space W formed as the direct sum of two spaces Y and U ($W = Y \oplus U$). The natural projection of W onto Y and U leads to the following:

$$\pi_Y: \begin{bmatrix} y \\ u \end{bmatrix} \mapsto y \quad \left(\begin{bmatrix} y \\ u \end{bmatrix} \in W \right) \text{ and } \pi_U: \begin{bmatrix} y \\ u \end{bmatrix} \mapsto u \quad \left(\begin{bmatrix} y \\ u \end{bmatrix} \in W \right)$$

so that

$$\begin{aligned} \sigma Gz &= Fz \\ y &= H_y z \\ u &= H_u z \end{aligned}$$

The following statements hold:

- 1) y processes u if and only if

$$H_y[T^*(H, sG - F) \cap V^*(H_u, sG - F)] = \{0\}$$

- 2) u is free if and only if

$$H_u[V^*(sG - F) \cap T^*(H_u, sG - F)] = U$$

Where the following notation is used:

$$T^k(H, sG - F) = T^k(H, sG - F; \{0\})$$

$$V^k(H, sG - F) = V^k(H, sG - F; Z)$$

$$V^k(sG - F) = V^k(0, sG - F) = V^k(sG - F, 0)$$

and

$$T^k(H, sG - F; T_0)(k \geq 0) \text{ is defined by}$$

$$T^0 = T_0, \quad T^{k+1} = G^{-1}F[T^k \cap \ker H]$$

$$V^k(H, sG - F; T_0)(k \geq 0) \text{ is defined by}$$

$$V^0 = V_0, \quad V^{k+1} = F^{-1}GV^k \cap \ker H$$

with the limit of T^k as $k \rightarrow \infty$ denoted as T^* .

Note that with this notation the controllable subspace for a standard state-space system $\dot{x} = Ax + Bu$, $y = Cx$ is $T^*(sI - A, B)$ which is the subspaces spanned by the columns of the controllability matrix $P = (B, AB, \dots, A^{n-1}B)$ [17].

Therefore, once the identification is accomplished, the experimental design should include a check of the subspaces resulting from these definitions to determine if a true input-output relationship has been found.

6. REFERENCES

1. Press, et al, *Numerical Recipes*, Cambridge University Press, 1986.
2. A. Gelb, *Applied Optimal Estimation*, The M.I.T. Press, Cambridge, 1974.
3. L. Maciejowski, *Multivariable Feedback Control*, Addison Wesley, Workingham England, 1989.
4. J. C. Willems, "Paradigms and Puzzles in the Theory of Dynamical Systems," *IEEE Trans. on Automat. Contr.*, vol 36, no. 3, pp 259-294, March 1991.
5. M. Kuijper and J. M. Schumacher, "Input-Output Structure of Linear Differential / Algebraic Systems," *IEEE Trans. on Automat. Contr.*, vol 38, no. 3, pp 404-414, March 1993.
6. L. Ljung, *System Identification: Theory for the User*, Prentice-Hall, NJ, 1987.
7. D. C. Montgomery, *Design and Analysis of Experiments*, John Wiley & Sons, NY, 1991.
8. L. Sachs, *Applied Statistics*, Springer-Verlag, New York, 1984.
9. Kendall, Stuart, and Ord, *Kendall's Advanced Theory of Statistics*, Volume 1, Distribution Theory, Oxford University Press, 1987.
10. G. Box, W. Hunter, and J Stuart, *Statistics for Experimenters*, John Wiley & Sons, New York, 1978
11. H. Cramer, *Mathematical Methods of Statistics*, Princeton University Press, 1963
12. L. Breiman and J.H. Friedman, "Estimating Optimal Transformations for Multiple Regression and Correlation," *Journal of the American Statistical Association*, Vol. 80, No. 391, pp 580-598, September 1985.
13. D. Belsley, E. Kuh, R. Welsch, "Regression Diagnostics, Identifying Influential Data and Sources of Collinearity," John Wiley & Sons, New York, 1980.
14. Franklin, Powell, Workman, *Digital Control of Dynamic Systems*, 2nd Edition, Addison-Wesley Publishing Co., Reading, MA, 1990.
15. R. E. Maine and K.W. Iliff, "Formulation and Implementation of a Practical Algorithm for Parameter Estimation with Process and Measurement Noise," *SIAM J. APPL. Math*, Vol. 41, No. 3, December 1981.
16. G. Goodwin, "Experiment Design for System Identification," *Encyclopedia of Systems and Control*, (M. Singh, ed.), Pergamon Press, Oxford, 1987.
17. H. Kwakernaak and R. Sivan, *Linear Optimal Control Systems*, Wiley-Interscience, 1972.

CHAPTER 10

METAMODELING COMBAT SIMULATIONS

1. TABLE OF CONTENTS

Metamodeling Combat Simulations	10-1
1. Table of Contents	10-1
2. Introduction	10-2
2.1. General	10-2
2.2. Simulation Requirements for Metamodeling Combat Simulations	10-3
2.3. Principles	10-4
2.4. Objective	10-4
3. General Approach	10-5
3.1. Problem Definition	10-6
3.1.1. Metamodel Purpose	10-6
3.1.2. Simulation Characteristics	10-6
3.2. Metamodeling Process	10-7
3.2.1. System Description	10-8
3.2.2. System Class	10-8
3.2.3. Metamodel Structure	10-9
3.2.4. Identification Methodology	10-9
3.3. Experimental Design	10-10
3.4. Verify the Metamodel	10-11
3.5. Summary	10-11
4. Metamodeling Procedures	10-12
4.1. Define the Problem	10-12
4.1.1. Metamodel Purpose	10-12
4.1.2. Simulation Characteristics	10-13
4.2. Select the Metamodel	10-16
4.2.1. System Description	10-17
4.2.2. System Class	10-18
4.2.3. Metamodel Structure	10-19
4.2.4. Identification Methodology	10-21
4.3. Experimental Design	10-25
4.4. Fit the Metamodel	10-28
4.5. Verify the Metamodel	10-28
4.5.1. Performance Analysis	10-29
4.5.2. Model Accuracy	10-29
4.5.3. Model Uncertainty	10-29
4.5.4. Model Adequacy	10-29
5. Procedural Notes	10-30
7. References	10-31

2. INTRODUCTION

2.1. General

With respect to Metamodeling Combat Simulations, the systems we are trying to identify are complex, nonlinear, time-varying discrete event systems. In general, for this case, the predictor function is a nonlinear function of past observations and there are too many possibilities for unstructured "black box" models. **Knowledge of the nonlinearities must be built into the model [1].**

Fortunately, in this case, we have explicit knowledge of the nature and characteristics of the simulation model that we are going to metamodel. Given this information, we can build the nonlinearities into the structure of the metamodel and provide the capability to generate a reduced-order approximation of the original model. This fact makes metamodeling as a method of model abstraction feasible.

Care must be taken in the setup of the metamodeling problem. **The experimental design must provide input-output sequences that correctly represent the system structure.** Unfortunately, determination that the data contains the correct representation of the system structure cannot be made before the generation of the metamodel. Only when we have the metamodel can we validate the data by identifying the probability of the data set given the parameters as the likelihood of the parameters given the data [2].

In addition to the problem setup and experimental design, the metamodel solution comes with limits of its own. Using the space spanned by the original model as the full order model, the metamodel is a reduced order approximation. This reduction inherently limits the span of the manifest (exogenous) variables associated with the behavior (input or output - if such a map exists). **Consequently, the behaviors allowed by the metamodel will exist within a subspace of the original model.** This subspace must be carefully analyzed.

Assuming that an input-output map exists for the model, **input values will be restricted to a domain** within which the metamodel will be applicable. Outside of this hypersurface, application of the metamodel may provide numbers but will not generate an output that is representative of the modeled system. Also, assuming appropriate inputs, **the output of the metamodel can only be guaranteed to be approximately correct.** Given a known system, every projection of that system into a subspace will reduce the information content of the observed behavior. The only exception to this is the situation where the kernel of the projection coincides with the null space of the behavior. Therefore, as a projection, the metamodel will not contain all of the detail of the original model. Output error bounds, that are a function of both the metamodel and the input, must be determined.

Recent advances in system theory have shown that much can be gained by using logic-based switching strategies [3]. Since fidelity, domain, and range are always tradeoffs in the approximation, it may be appropriate to define multiple metamodels over smaller

regions with higher fidelity to match a larger region of interest. Switching between these models will be a function of the domain of interest and desired fidelity.

2.2. Simulation Requirements for Metamodeling Combat Simulations

In Chapter 3 we defined system theoretic requirements for metamodeling simulations. We summarize these requirements here.

Military engagement simulations usually are defined to represent real-world events that have a beginning and an end. **Given that the simulation terminates naturally, results for complete systems can be applied** to the metamodeling problem since the system behavior is restricted to a finite dimensional sequence.

In general, the axiom of state is assumed because the simulation is set up in such a way that the initial conditions contain sufficient information about the past so as to determine future autonomous behavior. Also, an input-output structure with causality is assumed and evident in the presence of input and output files. These assumptions allow the application of Markov models even for the stochastic representation.

This only leaves the question of the information available in the data. Is the data sufficient? The answer comes from the definition of a dynamical system and must consider the behavior that is to be modeled, the representation selected/desired, and the data. For a stochastic system with multiple realizations, the ensemble of trajectories must span the space. Any single trajectory, for a stochastic or deterministic system, must span both the input and output space and be sufficiently long so that the state transition probabilities also span the allowable probability space and the distribution of these probabilities are the same as the underlying system. This condition can be assumed if the simulation reaches equilibrium. In this case, additional run time does not change the state of the simulation.

If the simulation does not reach equilibrium, there may still be adequate information in the data. This condition, however, cannot be verified without further testing.

In summary, assuming that the underlying system modeled by the simulation is well behaved (Markovian, complete with respect to the modeled behavior), the following is required to metamodel combat simulations:

1. The data must include the behavior we are trying to model.
2. The latent variables that define the behavior must be observable.
3. The input must be persistently exciting so that the effects of the latent variables are observed.
4. For a stochastic system, the ensemble of trajectories must span the space.

5. Any single trajectory must span both the input and output space and be sufficiently long so that the state transition probabilities also span the allowable probability space and the distribution of these probabilities are the same as the underlying system.

2.3. Principles

Physical insight is more important than anything else.

Use all available prior knowledge to reduce the uncertainty in the model estimates [4]. By examining the simulation carefully, we can use the structure and algorithms in the simulation to understand the implementation of the process being simulated. This understanding will help to define the metamodel structure.

In the general case of inverse modeling, the system is not known. However, if the dynamics of the system are available (as in a simulation) or can be assumed, it is possible to determine the number of processes that are present in the interconnected system.

During the inverse modeling process, it is imperative to model only one Markov system. Otherwise, behaviors associated with both processes will be aliased preventing the identification of either.

2.4. Objective

For our framework, we have defined a model class \mathcal{M} with elements $M = (U, B)$ where $B_f \subseteq U$ is the behavior of M . From an experiment we obtain data from measurements. Different realizations of the attributes of the phenomenon may give result in the same data, or they may lead to different observed data caused by the interference of other phenomenon or latent variables.

Multiple metamodels can be derived from the same data, our objective is to find the Most Powerful Unfalsified Model (MPUM). The more a model forbids, the better it is. A model is unfalsified by the data if $D \in U$ and $D \in B$. A model (U, B_1) is more powerful than (U, B_2) if $B_1 \subseteq B_2$. A model is the MPUM based on the data \mathbf{D} if: (1) $M \in \mathcal{M}$; (2) \mathcal{M} is unfalsified by \mathbf{D} ; and (3) M is more powerful than any other model satisfying (1) and (2). The MPUM may not exist. If the MPUM does exist, it is unique.

Then, given the model class \mathcal{M} (which is usually defined by the definition of the problem) and a data set \mathbf{D} , we present a metamodeling procedure that begins with the data and results in the collection of all subsets of $\mathcal{M} \Rightarrow$ the map $\mathbf{P}: \mathbf{D} \rightarrow \mathcal{M}$.

3. GENERAL APPROACH

In Chapter 3 we introduced a framework for the application of system identification techniques to develop suitable metamodels for tactical combat simulations used by the Department of Defense. We filled in the framework with concrete definitions and identified specific issues associated with the representation of dynamical systems. Particular attention was given to the discussion of experimental design requirements for metamodeling tactical engagement (usually Discrete Event System - DES) simulations. We demonstrated this approach by outlining the development of Ito stochastic and output error metamodels for the "Tactical Electronic Reconnaissance Simulation Model" which are described in Volume II, Chapter 2 and 3 and in [5,6].

Development of these metamodels followed the standard metamodeling procedure defined in [7] and introduced in Chapter 2. In this procedure, the first eight steps of the metamodeling procedure provide the prior knowledge or metamodel requirements that define the problem. The remaining steps define the experimental setup, the model structure, the method of identification, and validity measures used to develop and verify the metamodel.

Although our framework was consistent with the standard metamodeling procedures, the development of the metamodel required too many decisions to determine the model structure, method of identification, and identification criteria. Each decision was a complex function of *a priori* information and prior selections.

This chapter presents a new approach to support the development of metamodels. A new taxonomy of system representations (Chapter 5) and methods (Chapters 6 and 7) to generate the metamodel allows the separation of the metamodeling process into a set of sequential decisions based on *a priori* information.

This approach streamlines the development of techniques for metamodeling simulations by separating the procedure into two general areas. The first eight became the foundation for the **problem definition**, the remaining steps were grouped in an iterative scheme as the **metamodeling process**. Table 10.3.1 provides an overview of this structure.

Table 10.3.1 Metamodeling Approach.

MAJOR AREA	OBJECTIVE	DECISION/ACTION
Problem Definition	Metamodel purpose	Scope Use
	Simulation characteristics	External characteristics Internal characteristics
Metamodeling Process	Select a system representation and identification methodology	System description
		System class
		Metamodel structure
		Identification methodology
	Generate and verify the metamodel	Select an experimental design Gather data
		Fit the metamodel
		Verify the metamodel

3.1. Problem Definition

Recall that we defined a metamodeling problem as the direct sum of the metamodel requirements and the model (simulation). This means that the same simulation could be part of two different metamodeling problems if the requirements were different. Or conversely, the same set of requirements applied to two different (nonsimilar) simulations also leads to two different metamodeling problems.

Consequently, to define the problem we must consider both elements of the direct sum -- the purpose of the metamodel and the simulation characteristics.

3.1.1. Metamodel Purpose

As mathematical relationships, metamodels can be developed to support two general purposes:

1. Analysis
2. Hierarchical simulation

First, a metamodel can be used for analysis. In this case, the metamodel becomes an independent structure that is used to understand and extract information from the model.

Secondly, a metamodel can be used to support hierarchical simulation and model reuse. In this case, the metamodel is used in conjunction with (coupled to) other simulations or simulation elements to answer larger questions that are not supported within the structure of the modeled simulation. This purpose supports simulation based on a hierarchical representation. Using metamodels for this purpose is a two-step process. First a metamodel of a simulation (or component) is generated to develop more abstract simulation models. Then, once developed, these modules can be used to couple these metamodels (modules) to other simulations or metamodels to represent a more complex system.

Table 4.3.2 listed the scope and uses of analytical and simulation metamodels.

The selection of scope and use defines the metamodel purpose and provides clear boundary conditions for follow-on selections in steps 2 through 8.

3.1.2. Simulation Characteristics

We have discussed the purpose of the metamodel. Since all of the remaining problem definition decisions are a function (direct sum) of both the metamodel requirements and the simulation that is to be modeled, we concentrate on the aggregate space of simulation characteristics. Research has suggested that both a general (external) description of the simulation or model as well as further detail on the (internal) process structure of the internal components is required [5,6].

The classification defined by the "SIMTAX, A Taxonomy for Warfare Simulation" and presented in Chapter 4 is completely adequate for the external description. It is a

descriptive framework designed to guide the development, acquisition, and use of warfare models and provides the basis for classifying objects for identification, retrieval, and research purposes.

Selection of a metamodel structure, however, requires detailed information not contained in the simulation and model catalogues. To provide a link between the more general taxonomy outlined above and specific metamodeling techniques, a more detailed internal taxonomy was appended to the SIMTAX. The purpose of this additional detail is to describe the structure of the simulation in terms of system theoretic definitions common to control engineering.

Figure 4.4.1 depicted the model of a continuous system with a sampled measurement and the text with the figure explained the concept. In development of a metamodel, we try to isolate and identify each of the individual elements in this model. Consequently, we must be able to characterize the type of processing that takes place in each of the blocks.

Formulating the metamodeling problem in this manner is important for two reasons. First, it is usually not possible to simultaneously identify more than one process. If the processes are independent, a rank deficiency in the uncoupled equations causes numerical difficulties. If the process are dependent, behaviors associated with both processes will be combined preventing the identification of either.

If one is successful in simultaneously identifying multiple process, performance of the resulting metamodel is usually poor. Unless the model structure and order accurately accommodates both systems, the minimization process used for identification will generate a system of equations that represents the combined behavior but neither system well.

Categories and selections for these categories that were used to provide the additional detail on the internal structure were discussed in Chapter 4 and summarized in Table 4.4.16. To determine this additional information we review the simulation. We will determine inputs, latent variables, and outputs and identify the relationships among the variables. From the calculations, we will also identify the number and type of processes (systems) contained in the simulation and determine the variables that are needed to identify each system.

3.2. Metamodeling Process

At this point, we have determined the purpose of the metamodel. In the definition of this purpose, we have identified the input and response of interest and have determined the important characteristics of these data. Also for this purpose, we have defined the region of interest, selected validity measures, and specified the required validity.

Now we discuss decisions associated with "Step 9: Postulate a metamodel." The completion of this step requires a number of interrelated selections. However, the combination of model selection, error criterion, identification technique, and numerical methods leads to an overwhelming myriad of "identification methods."

In fact, there seem to be as many system identification methods as there are inverse problems. Many specific identification and statistical methods have been developed to accommodate the differences in model structures, data length, and measurement error statistics, etc. The literature contains considerable discussion on particular methods with very little discussion on the relationship of these techniques to each other or to a general methodology. The result is a confusing array of unconnected methods with little or no guidance on the application of the techniques to general classes of problems.

Since we are looking for procedures to handle general metamodeling problems, we discuss these methods as elements of a more general structure. We have reduced these decisions to four. The first two decisions concern the system description and class. The next decision defines the structure of the metamodel, while the last selection provides the identification methodology.

In reality, all "real world" systems are complex, large scale interconnections of continuous-discrete, nonlinear, infinite-dimensional components. We will approximate these systems with lumped parameter, parametric, finite dimensional models.

A model representation (set) is defined by the **system description**, **class**, **structure**, and **identification methodology**. Multiple model sets are available and the performance of the metamodel will be limited by the match between the metamodel set and actual system that generated the behavior.

3.2.1. System Description

In the definition of the system description, the first selection concerns the system type that will define the allowed behavior of the models. Here, the most basic questions must be addressed. How are the parameters described? Is the representation going to include dynamics or will it be static? Will the model contain latent variables? If it dynamic, is it time invariant or time varying?

Is the algebraic structure linear or nonlinear? Are disturbances, noise, and randomness accommodated? Is the system defined as continuous, discrete, continuous-discrete, or as a discrete event system? Table 5.3.2 outlined the possible selections that define the system description.

3.2.2. System Class

In addition to the system description, the class of representation is also needed to define the overall model set. This class is defined by the interaction of the variables and the representation. Table 5.3.3 provided a list of the general system classes and the possible form of the representations [8,9].

3.2.3. Metamodel Structure

Once the system description and class have been determined, the next decision is selection of the model structure to use in describing the response of the system to the inputs (possibly including latent variables). There are many options here and this selection generates much of the complexity in system identification.

A model structure M is defined as a differentiable mapping from a connected open subset D_m of R^d to a model set $M(\theta)$, such that the gradients of the predictor functions are stable.

We define two general model structures, **predictor models** and **probabilistic models**. A predictor model only defines the predictor equation(s). Predictor models are models that specify the elements of the transfer function in terms of some parameter set. The models generated from these structures are deterministic in nature. Predictor models, however, do allow for the prediction or measurement error. And since the coefficients were generated via a minimization of some error criterion with assumed statistics, the coefficients will be random variables with an error distribution. Since the estimates are functions of these random variables, this distribution can be used to compute error bounds of the estimate.

A probabilistic model accommodates the fact that many systems are subject to known disturbances that are not (or cannot be) completely categorized. The statistics of the noises and disturbances are included as random variables. Probabilistic models supplement the parametric description with a description of the density function (or moments) of the noise (disturbance) that acts on the system. The variables of the system being identified become functions of random variables. In these situations, different realizations of an experiment (simulation run) may not produce exactly the same results. Consequently, the output of a probabilistic model is the conditional expected value and probability density functions (CPDF) of the variables.

Depending on the system class, either of these model structures can be expressed in one of three forms. They can be expressed as a polynomial, a matrix fraction, or in a state space form.

These two model structures were discussed in detail in Chapter 5.

3.2.4. Identification Methodology

As of this time, we have chosen the system description and class, and have selected a model structure that we will use for the identification. We now discuss techniques for generating the estimate. Consequently, this subsection covers the many methods available to support decisions associated with "Step 12: Fit the metamodel."

Parameter identification methods are used when the candidate model is to be defined by a set of parameters. Parameter estimation algorithms mentioned in the literature include

least squares, sequential weighted least squares, recursive generalized least squares, instrumental variables, recursive instrumental variables, the bootstrap method, sequential correlation, and recursive maximum likelihood estimation.

Most of the above techniques can be classified by two elements of the identification method: the **form of the identifier** and the **criterion of fit**. The form of the identifier defines the "experimental setup" (Equation Error Method, Output Error Method, Prediction Error Method) or the manner in which the estimates are generated and compared.

By criterion of fit, we mean the function or functional that is optimized to determine the parameter estimates. The criterion of fit establishes both the cost function and the method of its minimization. We consider three criterion: minimum mean square, maximum a posteriori (maximize the CPDF), and maximum likelihood (maximize the JPFD).

3.3. Experimental Design

The "experimental design" are the methods used to structure an experiment, test, or series of tests and the method selected depends on the metamodel representation. The purpose of the structure is to make purposeful changes in the input variables so that we may observe and identify the reasons for changes in the response.

The procedure we will follow differs from procedures outlined in Chapter 9 in that the selection of the model set, structure, and identification criterion has already been accomplished based on the characteristics of the simulation (or simulation component) that we are metamodeling. Here we concentrate on the selection of the input-output data and treatment of that data. Selection of run order, and a blocking or randomization procedure to structure the inputs, depends on the problem definition and system representation.

A. Input-Output Data

1. Select dependent signals (outputs)
2. Select driving signals to measure (inputs)
3. Select sampling interval
4. Select input characteristics (spectra)
5. Choose the number of samples to collect.

B. Obtain data

C. Treatment of Data

1. Regression Diagnostics
2. Deviations from equilibrium.
3. Subtract sample means.
4. Estimate offsets and drifts explicitly.
5. Difference the data.

3.4. Verify the Metamodel

Parameters are meaningless unless the mathematical model correctly describes the behaviors addressed. Consequently, it is advisable to perform a number of experiments to prove that the model correctly predicts the responses for different inputs. A scatter plot of parameters identified from different experiments provides the most credible indication of parameter accuracy [10].

In MMLE, the Hessian is computed from the inner product of the sensitivities so it cannot be singular unless a combination of the parameters has no effect on the estimated output. Failure of MMLE due to poor conditioning of the Hessian is a useful indicator of identifiability problems that must be resolved.

3.5. Summary

From the above discussion, we see that the metamodeling decisions flow directly from the problem to be addressed. The results of each decision flow directly to subsequent decisions.

4. METAMODELING PROCEDURES

The procedures that will be used follow the metamodeling steps given in [7] (shown in boldface) and the approach outlined above.

4.1. Define the Problem

This part of the procedure determines prior information for construction of the model. These steps must provide the general idea of the objective of the metamodel along with specific data to support construction of the metamodel.

The following procedure is general in nature. There are several situations that could be encountered: metamodeling of a single realization of a simulation; multiple realizations with different initial conditions; or a Monte-Carlo ensemble of the same initial conditions. Each of these may require some modification to the general procedure given below.

4.1.1. Metamodel Purpose

Review purpose of metamodel. Is this an analytical or simulation metamodel? If it is an analytical model what are the questions that are to be answered? What data are required to answer these questions? If it is a simulation metamodel, what are the inputs to the simulation or component? What outputs are required? Figure 10.4.1 charts the beginning of the process.

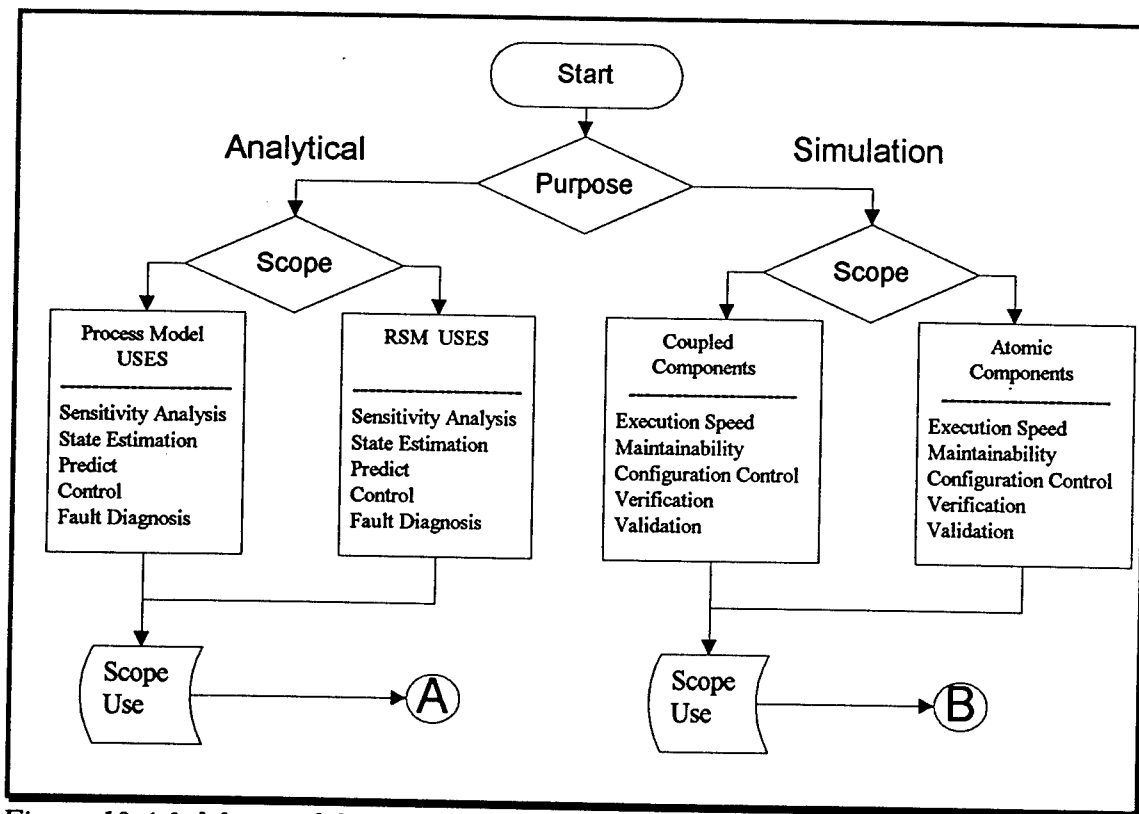


Figure 10.4.1 Metamodeling Approach

4.1.2. Simulation Characteristics

Having defined the purpose of the metamodel, we must now identify specific elements of the simulation that are to be metamodeled. We first consider the flow for analytical metamodels shown in Figure 10.4.2. The discussion above began with the external description and then appended an internal description. Since the internal description is required to understand the simulation and the external description is used in the metamodeling process, we complete the internal analysis first.

To begin, we review the simulation to determine the program flow. Identify the structure of the simulation, the processes that are simulated, where and how they are implemented. Identify the relationships among the variables. Defining the decision points will help in determining the number of simulated processes. For each separate process that is simulated, **identify input factors**. We must determine the input variables and determine the domain of the input, the manner and location of their storage, and how they are formatted.

Recall that the inputs we are discussing are not just the inputs to the overall simulation (which may only be initial conditions). They are the inputs to each of the processes which may be internal to the simulation. **Identify important input characteristics**. The input spectrum for each of the processes should be determined.

In the process of reviewing the simulation, we should also identify the latent variables that are not in the input or output. Determine where they are calculated, if/how they are modified or updated, and where and how they are stored. Parameters, both input and embedded, should also be catalogued.

Identify the response. Identify important response characteristics. Determine the calculations required for output variables and where these calculations are carried out. Determine the location of the output variables, their formatting, and the range of the outputs as defined by the calculations.

The final step of the process is to couple inputs and outputs available from the simulation to the metamodel purpose. The manner in which the simulation is going to be used to address that purpose must be clearly understood. The analytical requirements that defined the purpose of the metamodel may require variations in specific parameters or calculations involving combination of outputs (the result may or not be a direct output of the simulation).

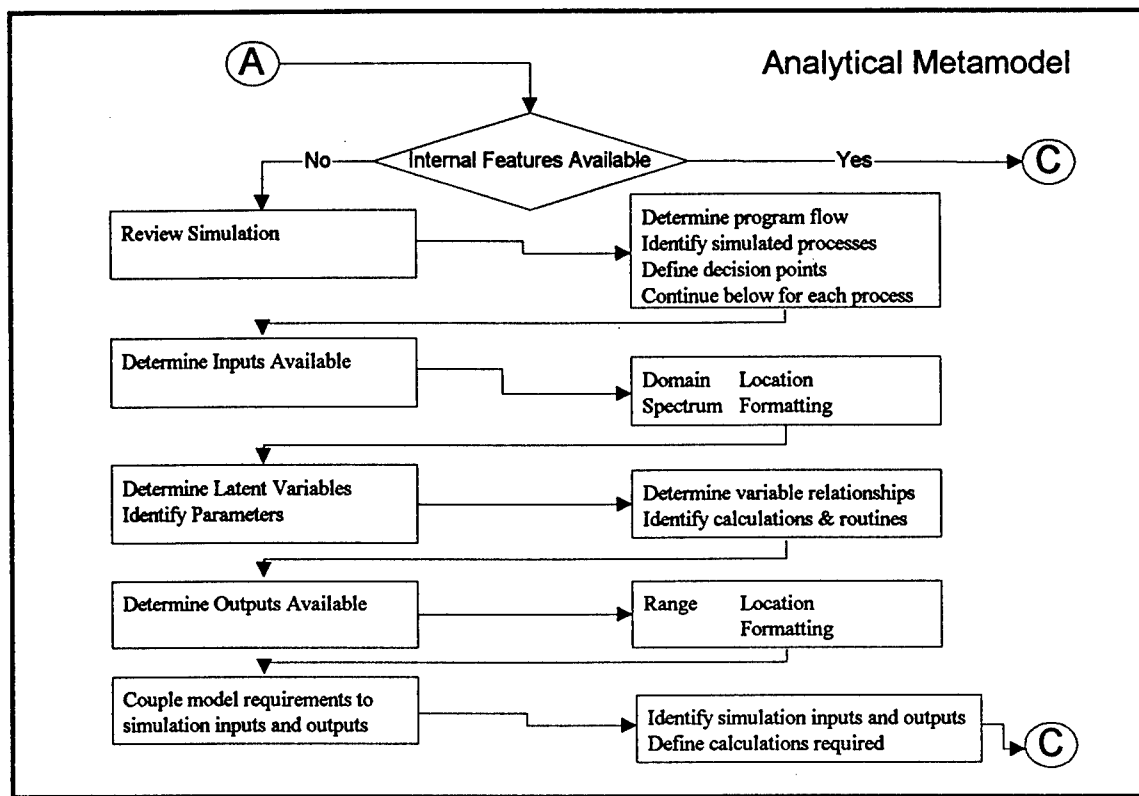


Figure 10.4.2 Analysis of a Simulation for an Analytical Metamodel

The initial analysis of a simulation for a simulation metamodel is similar to the analytical metamodel with the exception that the inputs and outputs of the metamodel are defined by the simulation. Also, in general, there are no options on how the inputs and outputs of the simulation will be used. All inputs to the simulation (or simulation component) and all outputs from the simulation (or component) must be accepted and provided. If the simulation or component is going to be used in a different application, subsets of the inputs and outputs could be extracted (with care) from the full set to support the new requirement.

Although all of the inputs and outputs must be explicitly considered, latent variables only need to be included in the metamodel to the extent that they support accuracy requirements. The influence of these additional, but unmodeled, parameters must be understood. Also, we must be concerned with the frequency of calculation and storage for latent and output variables.

The process for the analysis of a simulation metamodel is depicted in Figure 10.4.3.

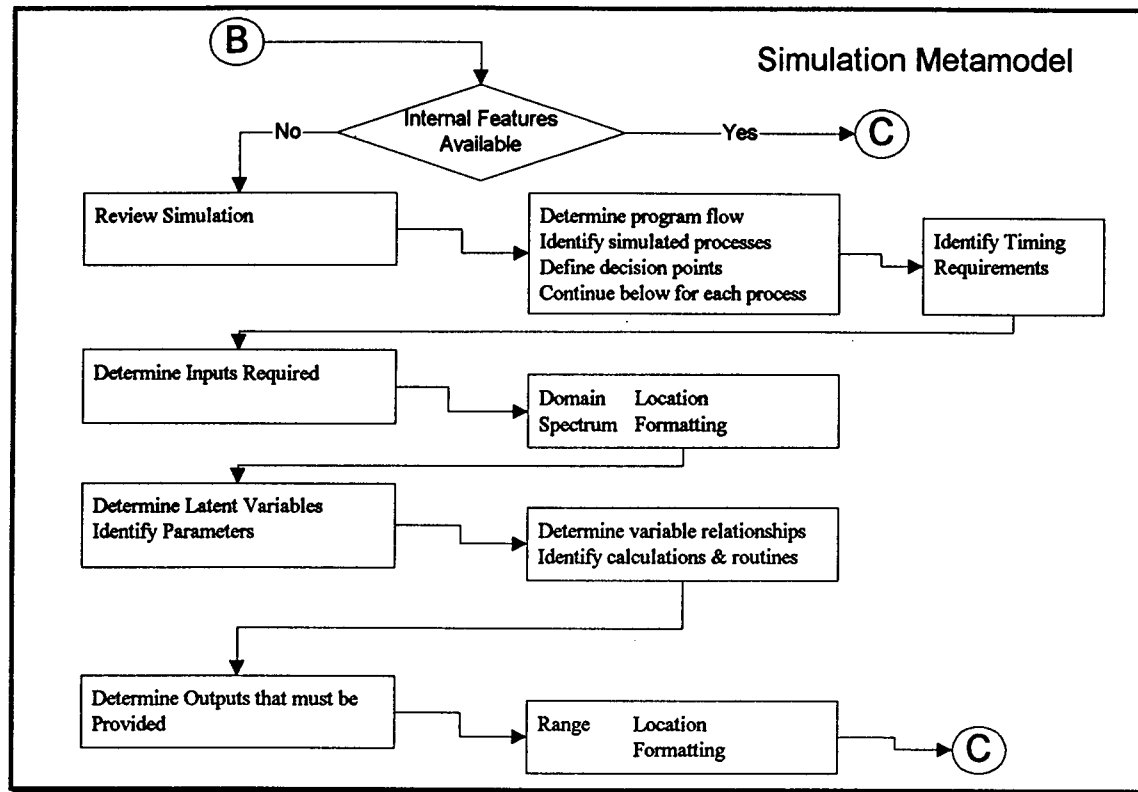


Figure 10.4.3 Analysis of a Simulation for an Simulation Metamodel

Once the internal operation is understood and an internal feature vector computed, we consider the external description (Figure 10.4.4). If available, we use the existing SIMTAX feature vector. Otherwise we load the simulation into the SIMTAX database and run the query (see Volume II, Chapter 5). Once both feature vectors are available we compare them to insure that they are consistent. We should also determine the relevancy of any prior metamodel data by objective and resolve any ambiguities that may exist.

With an understanding of the purpose of the metamodel and the simulation, we must now connect the simulation to the purpose. Here we need to evaluate the characteristics simulation with respect to the purpose of the metamodel. We complete the definition of the problem. We have identified the range and domain of the simulation. What is required for the metamodel? To answer this question, we **specify the experimental region**. We have determined the outputs, we now determine output accuracy required and the range of outputs that are of interest. Once we know the output range, what domain and structure of the inputs is required to get the range of output? Once the experimental region is defined, we use the purpose of the metamodel and the characteristics of the simulation to **select validity measures and specify required validity**.

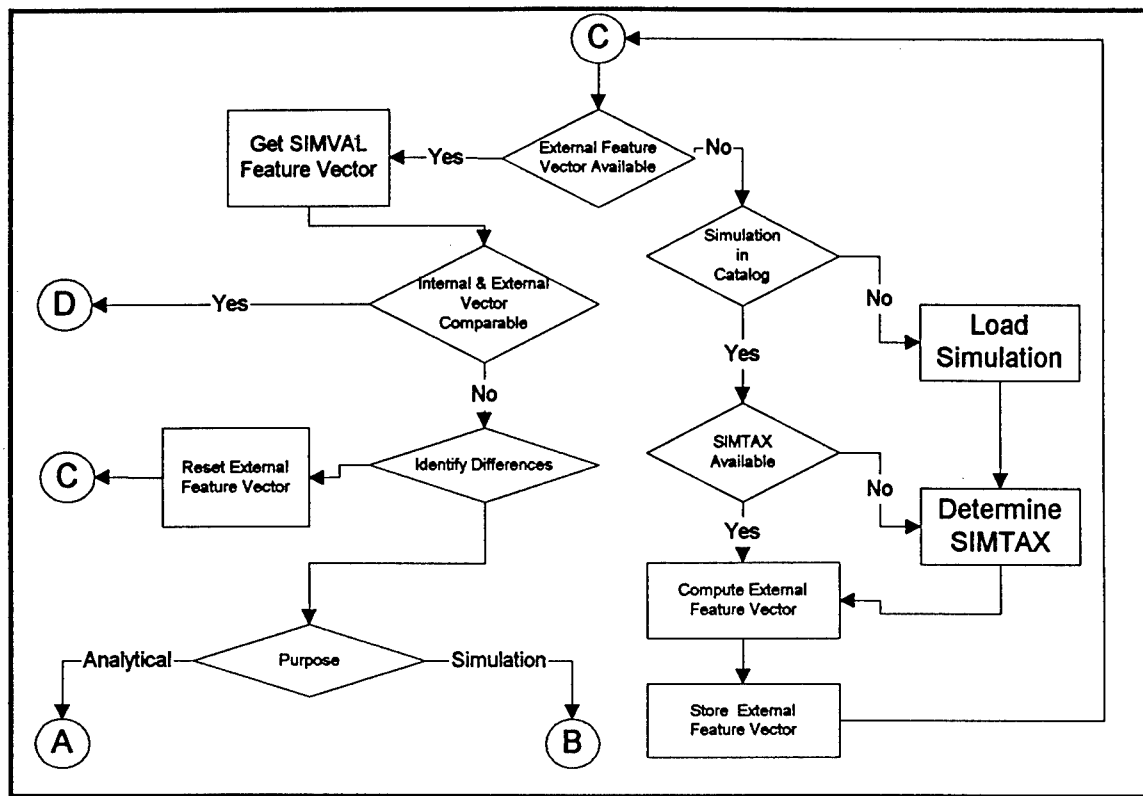


Figure 10.4.4 Analysis External Simulation Features.

With the initial definitions accomplished we should again insure that all of the objectives are consistent. If there are multiple objectives, or if the simulation includes multiple processes, combinations of metamodels or a hierarchical set of metamodels may be required to meet the requirements. Is new data needed from a simulation or does the data exist - if so, where? We should also identify special resources required.

4.2. Select the Metamodel

From our analysis of the simulation, we have defined the initial model, M_0 that gives accurate values of standard performance measures but is expensive because of internal complexity, many parameters, etc. The parameters of M_0 are P_0 . The set of performance quantities (validity measures) of M_0 that are of interest are Q_0 . To determine these quantities the model M_0 , with the set of parameters P_0 must be solved. To generate the metamodel, the analyst seeks an approximate (transformed) model M with parameters P , and the following properties [11]:

1. The parameters P are easy to derive from M
2. The solution process is fast
3. The performance metrics Q_0 are easily obtainable from M
4. The approximations in the transformations between M_0 and M do not introduce too much error.

The metamodeling process consists of the construction of the forward mapping F for specifying M and its parameters P from M_0 and P_0 . This involves homogeneity or aggregation assumptions. From M and P we identify the dynamical system. Then we evaluate the approximate model M to determine Q . If necessary we construct the reverse mapping R . Then we evaluate the error.

The basic process can be extended in many ways. The basic model can be transformed into a set of models, the combination of which approximates M_0 . The individual models are designated as (M_{oi}, P_{oi}) . Possibly, the process can be applied to the submodel to further refine the approximation. Additionally, the process may involve iteration of forward mapping function until the error is less than some allowable bound. This further refinement can be via recursion, where the model is expressed in terms of simpler versions of itself. This result is a hierarchy of models. The basic process for a specific model is outlined in Figure 10.4.5.

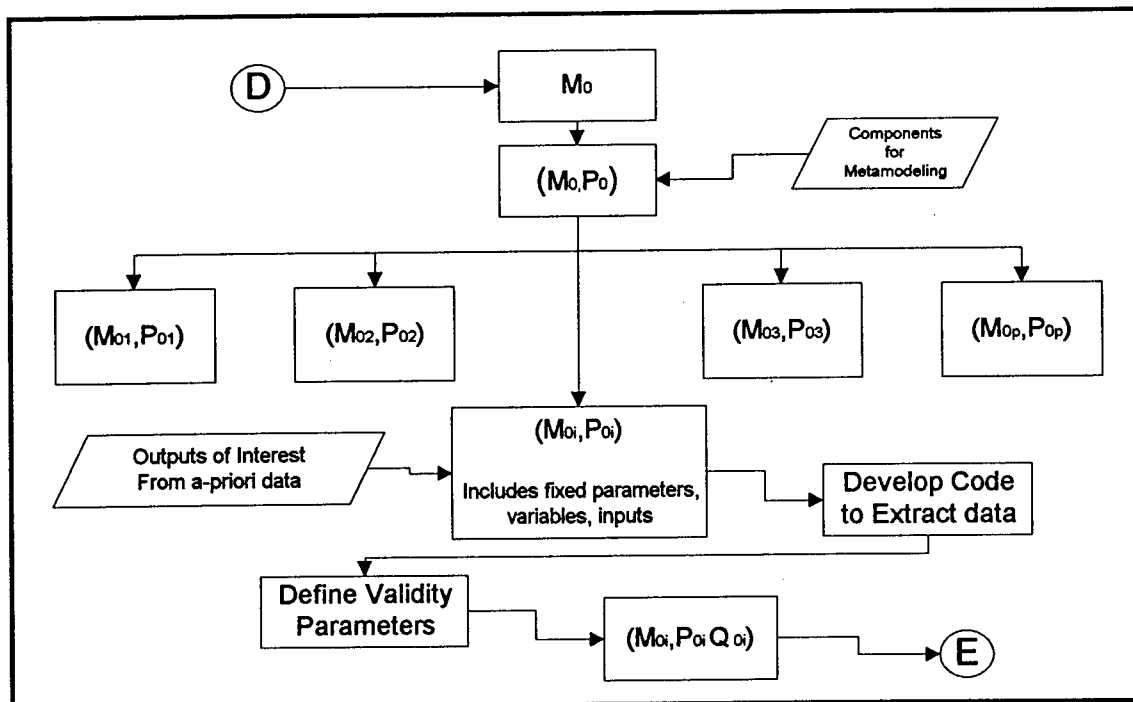


Figure 10.4.5 Selection of Metamodel Components

We have completed the initial definition of the metamodel. Now we are ready to postulate a metamodel based on: input-output response characteristics; experimental region dimension; and required validity. We separate these decisions into system description, system class, and metamodel structure.

4.2.1. System Description

Description of the system is accomplished by defining the system type, algebraic structure, randomness, time, and trajectory of the processes modeled by the simulation. This series of decisions is shown in Figure 10.4.6.

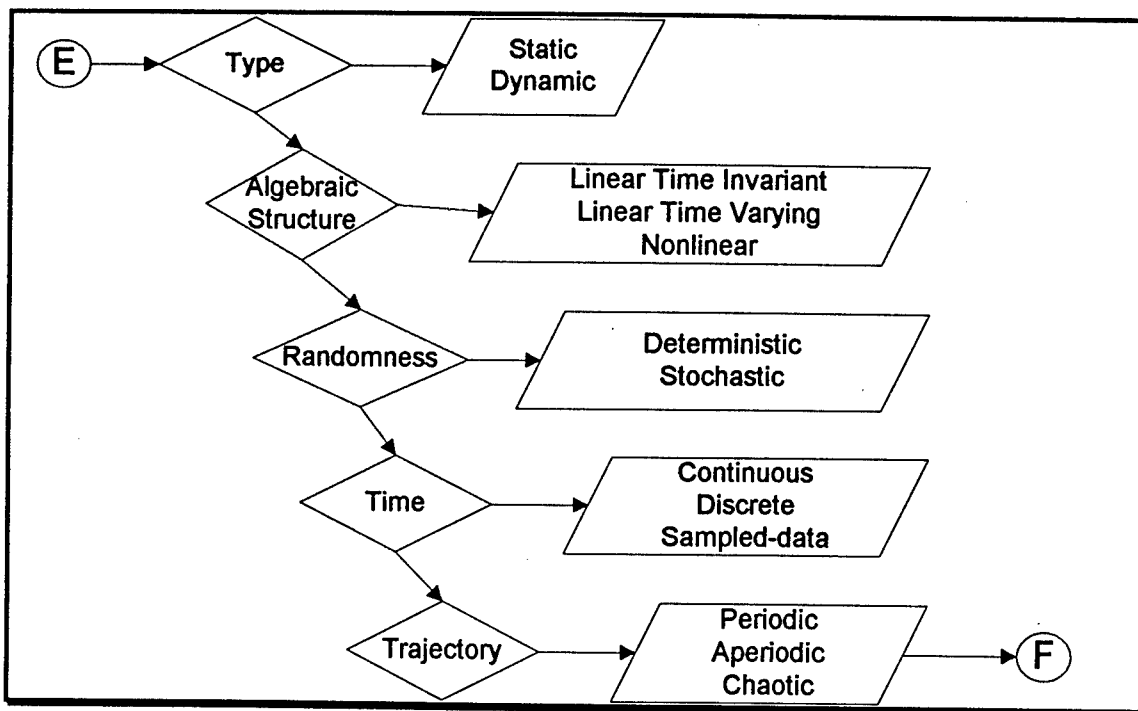


Figure 10.4.6 Determination of System Description.

4.2.2. System Class

The system class is a function of the number of inputs and outputs. A SISO system has a single input and a single output. A MISO system has a multiple inputs but a single output. A MIMO system is the most complex and has multiple inputs coupled to multiple outputs. This is shown in Figure 10.4.7.

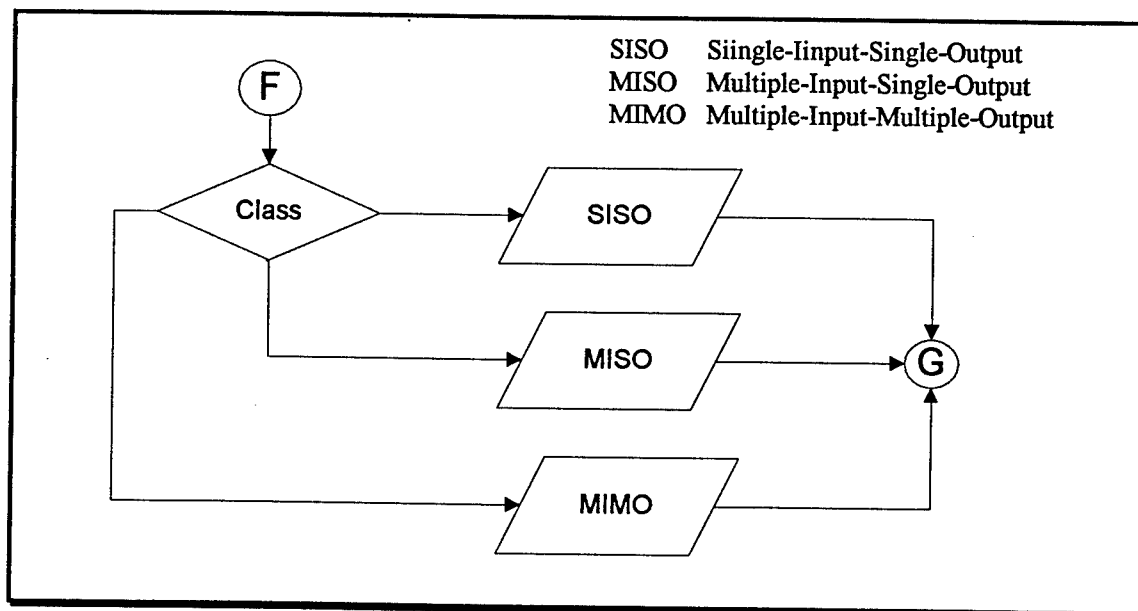


Figure 10.4.7 Determination of System Class.

4.2.3. Metamodel Structure

Based on prior information, the system description, and the system class, Select and define a model set from within which a model is to be found. The selection of the structure will define requirements for the experimental design.

Figure 10.4.8 shows the models available for static system models.

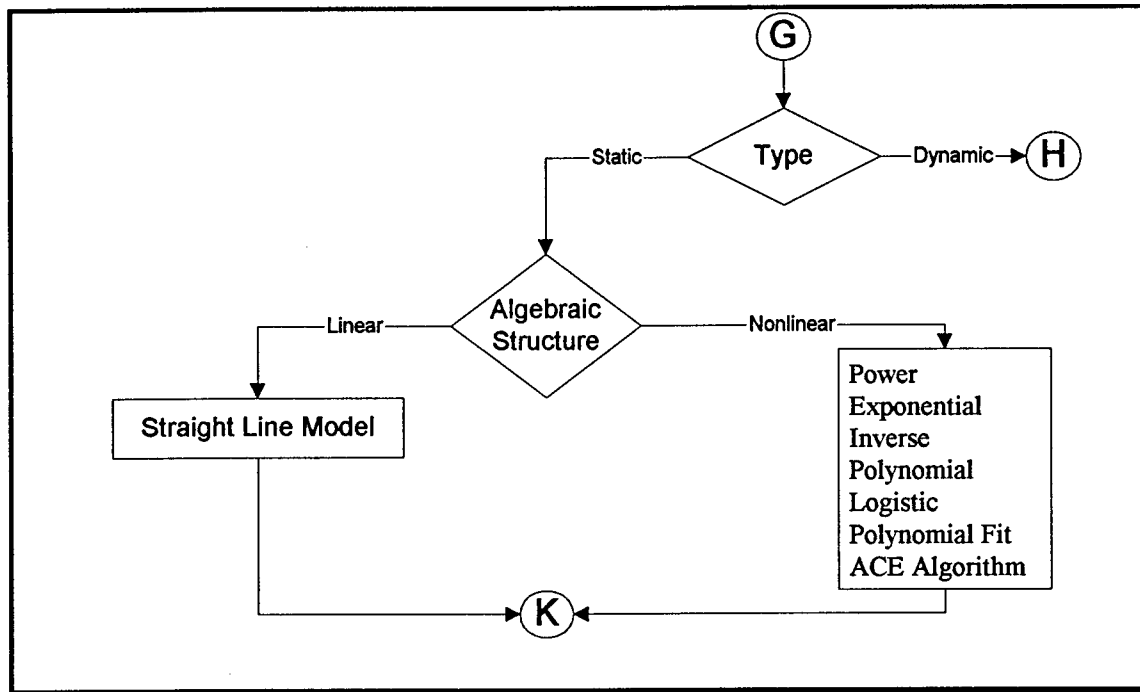


Figure 10.4.8 Metamodel Structure for Static Systems.

The model sets for linear time-invariant dynamic systems are shown in Figure 10.4.9. Since the theory for matrix fraction descriptions is not well developed, they are not included as a possibility for the linear time invariant MIMO case.

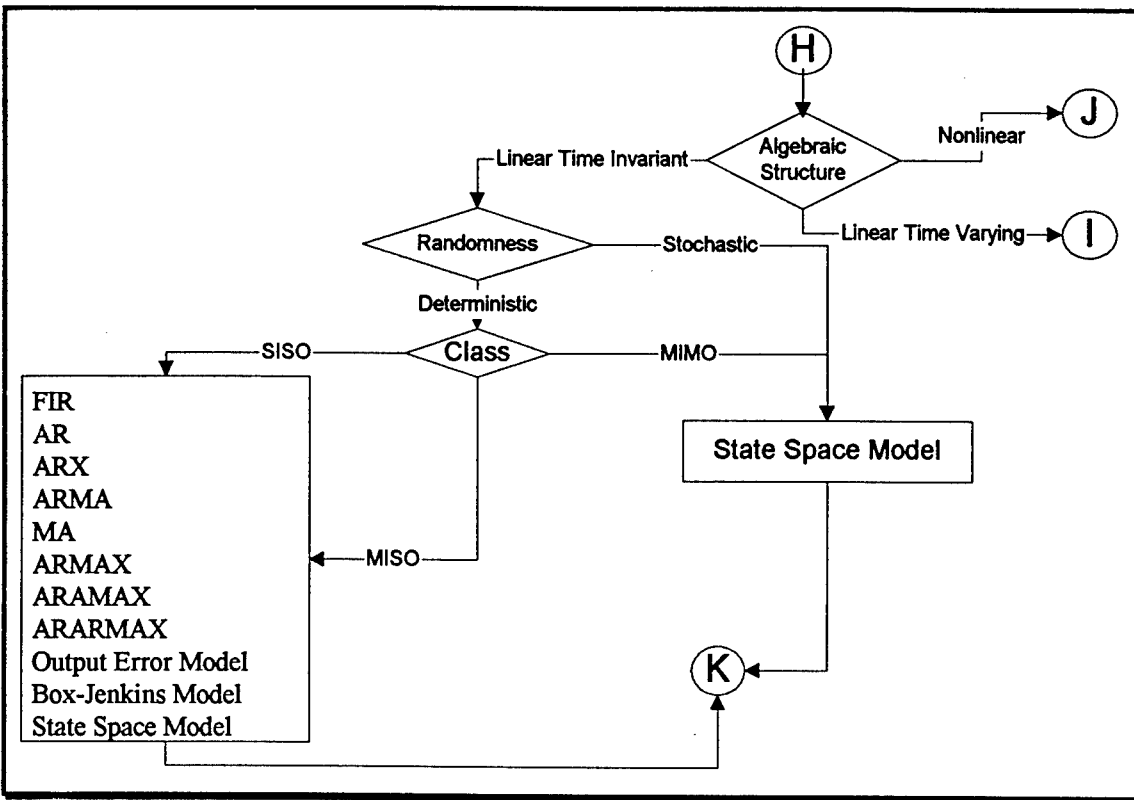


Figure 10.4.9 Metamodel Structure for Linear Time-Invariant Systems.

Linear time-varying dynamic models sets are shown in Figure 10.4.10. A selection not shown in the conversion of the time-varying system into a series of time-invariant systems that can use time-invariant model sets.

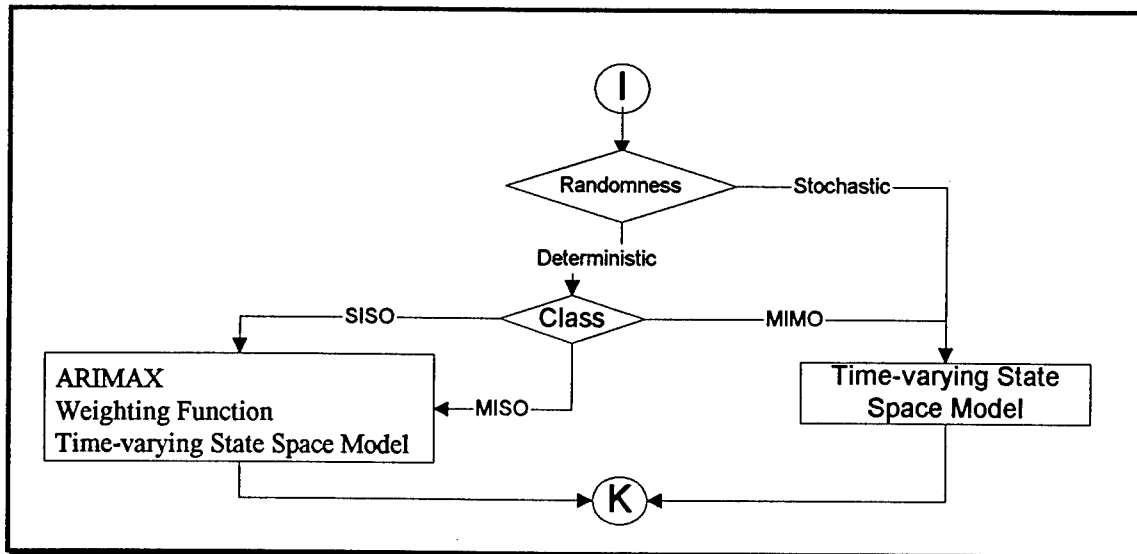


Figure 10.4.10 Metamodel Structure for Time Varying Systems.

Nonlinear models sets are shown in Figure 10.4.11. If linearization is selected then a time-varying or time-invariant model can be used.

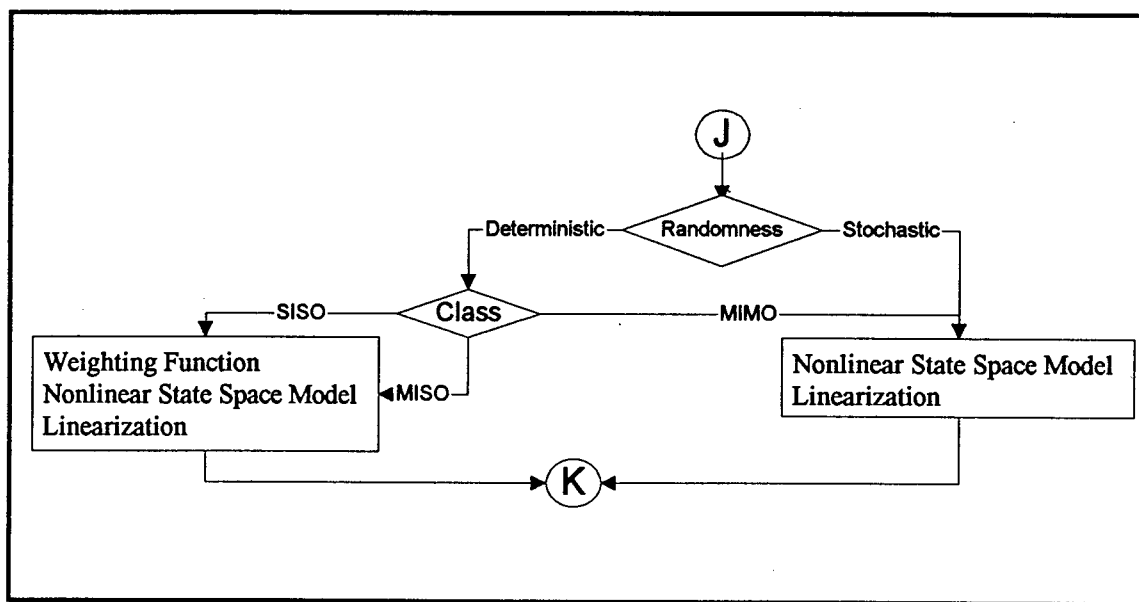


Figure 10.4.11 Metamodel Structure for Nonlinear Systems.

4.2.4. Identification Methodology

The identification methodology is classified by two elements of the identification method: the form of the identifier and the criterion of fit. The form of the identifier defines how the data are generated. The criterion of fit is the function or functional that is optimized to determine the parameter estimates. These selections are covered in Figure 10.4.12.

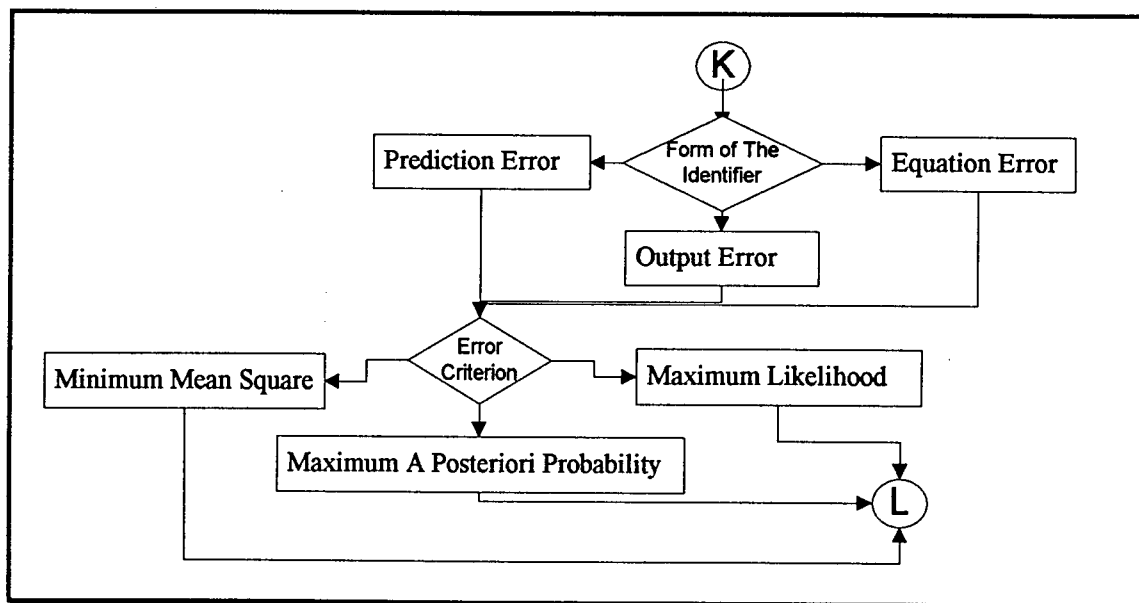


Figure 10.4.12 Selection of the Form of the Identifier and Error Criterion.

Given the definition of the problem, the metamodel structure, the form of the identifier, and error criterion set, we can now select the identification algorithm. Although covered here, implementation of the "Select ID Routine" is actually included under "Fit the Metamodel" because of the iterative nature of the process. After an initial selection, depending on the results, the method may have to be changed. The main "Select ID

Routine" is shown in Figure 10.4.13. Subroutines to breakout specific methods from the general categories follow. This selection process is based on research experience and is designed to provide the most robust solution. It is not the only path that will lead to a successful metamodel.

One of the difficulties with multivariable problems is the selection of the initial conditions. This research has shown that multivariable probabilistic models are best initialized by a correlation or prediction error method. The equation error or output error methods usually use a minimum mean square error (MMSE) criterion. A time varying predictor uses linear routines and can explicitly include the time varying behavior or breaks up the data into time-invariant sections.

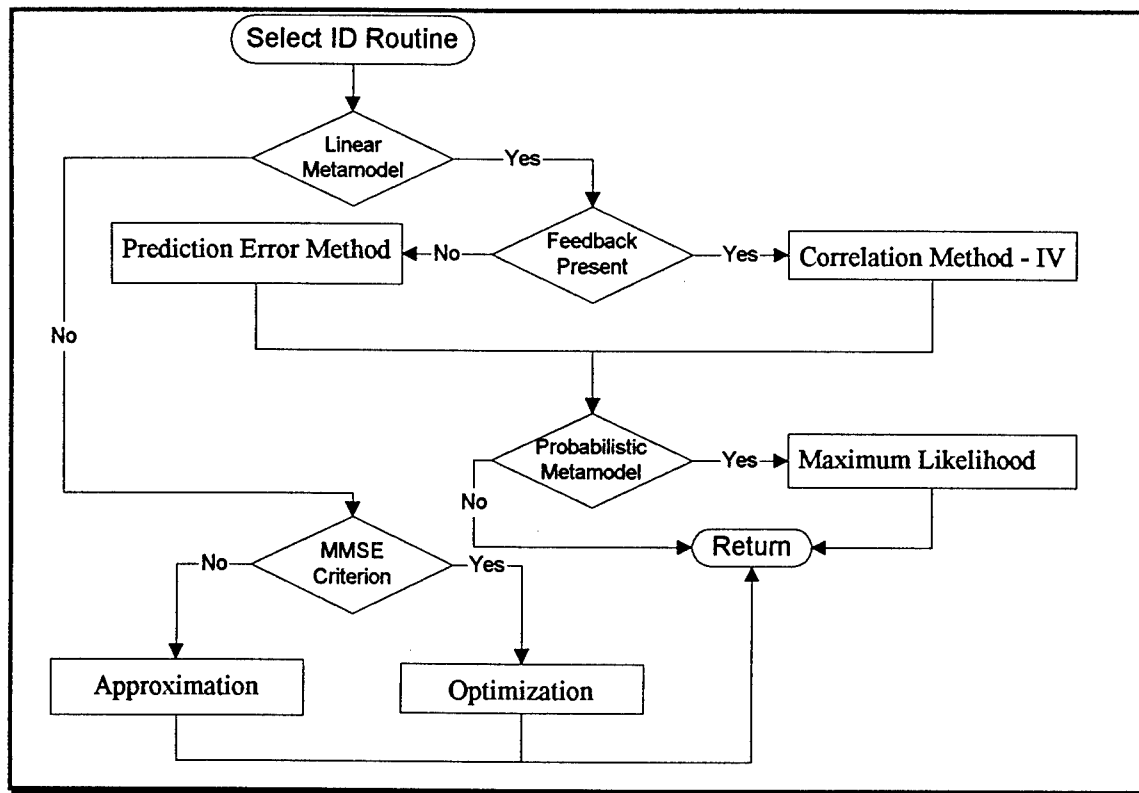


Figure 10.4.13 Main ID Method Selection Routine.

Many techniques can be formalized as special cases of the general multivariable prediction error method (PEM) and most problems can be solved with PEM. The path in Figure 10.4.14 is set up to provide the most efficient routine capable of generating a solution.

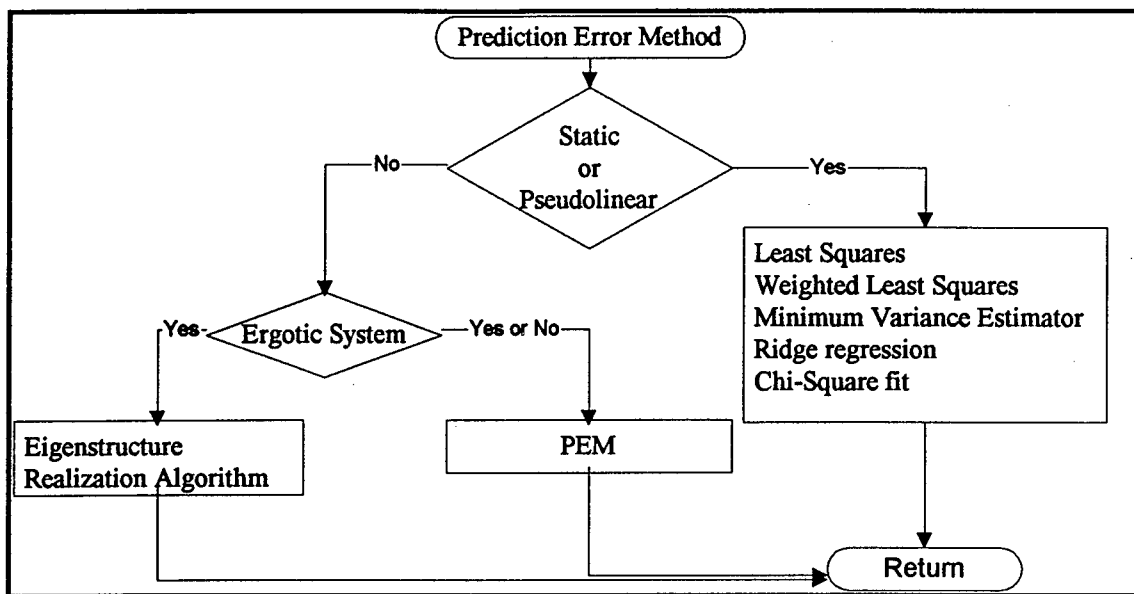


Figure 10.4.14 Selection of Prediction Error Methods.

Maximum likelihood methods are shown in Figure 10.4.15. The linear Ito stochastic model are optimized using ASA uses a MMSE cost function and should be used when the disturbances such that the "white noise approximation" to the Wiener integral does not hold. Although primarily linear methods, both the full scale estimator and MMLE can be modified for nonlinear propagation and measurements.

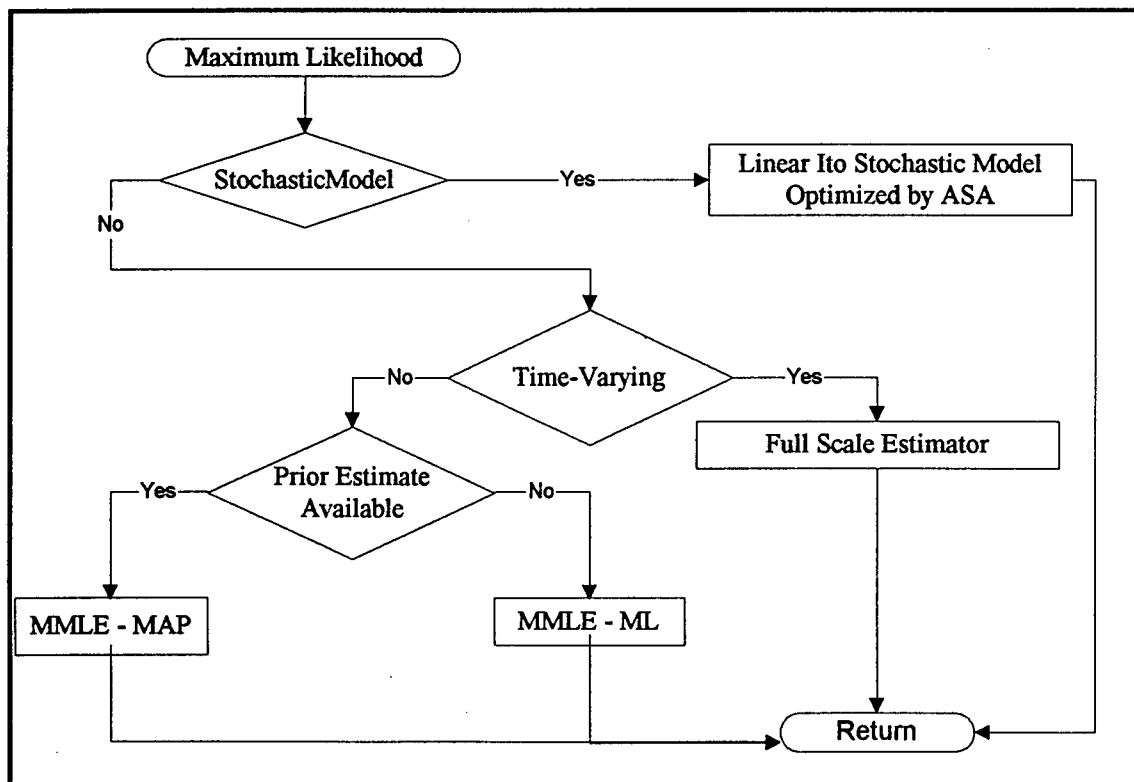


Figure 10.4.15 Selection of Maximum Likelihood Methods.

CVA is the most powerful of the approximation methods in Figure 10.4.16 and can accommodate full nonlinear systems. It does have a problem with an explosive increase in the order of the problem when multi-input systems are considered.

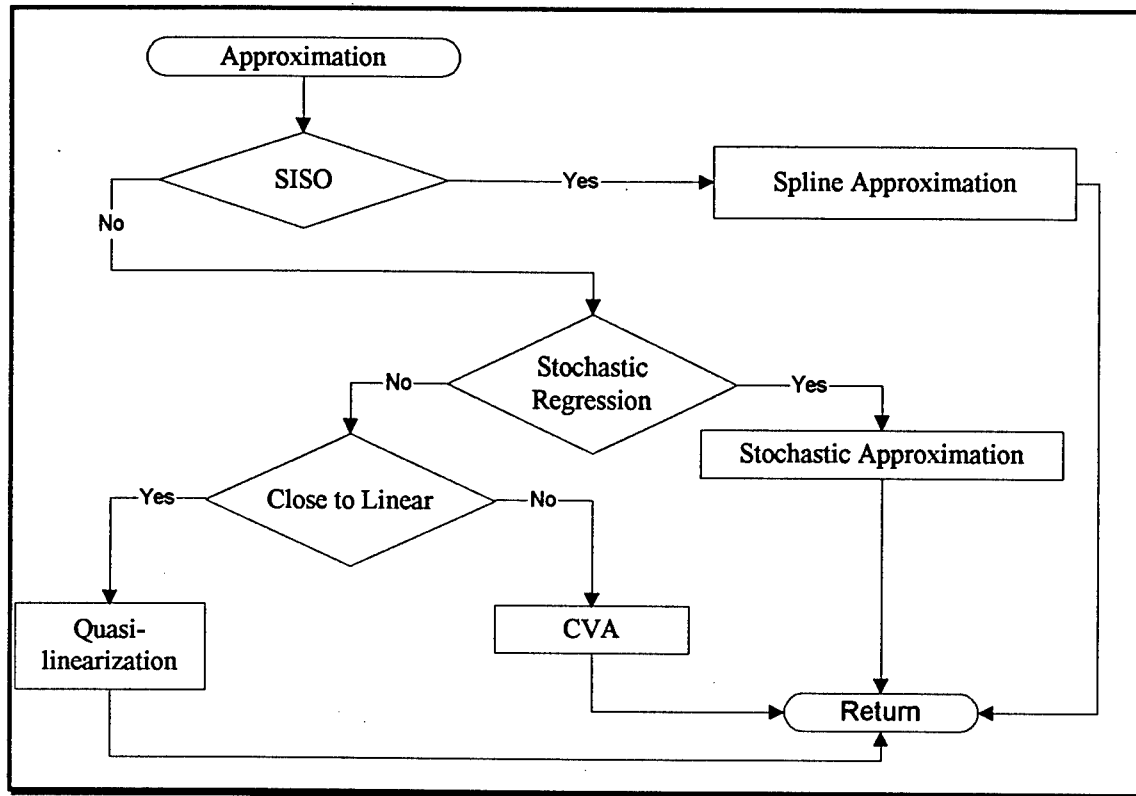


Figure 10.4.16 Selection of Approximation Methods.

The most general optimization technique is ASA. It can be used with simulation models or the nonlinear filters to minimize a cost function. If a probabilistic model is not required, pEST can handle general nonlinear time-varying systems. The optimization methods are shown in Figure 10.4.17.

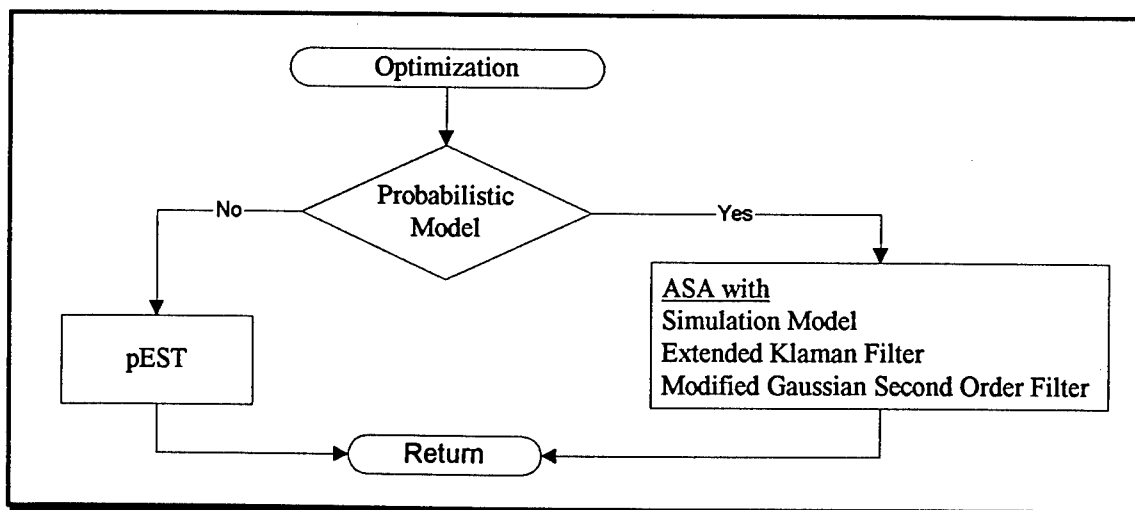


Figure 10.4.17 Selection of Optimization Methods.

4.3. Experimental Design

We have used the information provided for in the problem definition to select a model set and are now ready to **select an experimental design**. The type of design that we select depends on the structure of the metamodel and the data that will support the model. The first principle for the experimental design is to use all available prior knowledge to reduce the uncertainty in the estimate. Consequently, we want to partition the behaviors (systems) into components small enough so that we can measure the observable states. This partition is limited by the fact that we need to make sure that the behaviors we are trying to represent are complete.

The guidelines and methods for experimental design are given in Chapter 9. Our procedure to select the values of the input variables is shown in Figure 10.4.18.

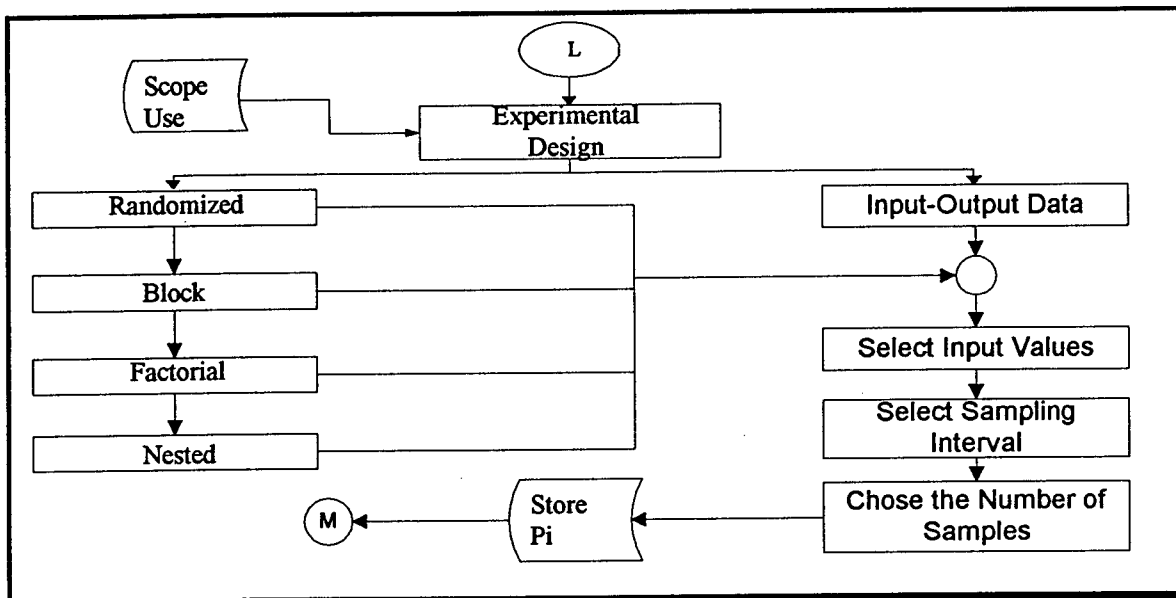


Figure 10.4.18 Selection of Simulation Input Values.

If purely statistical methods are used, classical "experimental design" is the process of designing the experiment so that appropriate data can be analyzed by statistical methods that require identically independently distributed (IID) random variables [12,13,14].

The three basic principles of experimental design are replication, randomization, and blocking. Designs for static metamodels using linear or pseudo linear regression follow:

1. Randomized Blocks
2. Latin Squares
3. Graeco-Latin Design
4. Incomplete Block Design
5. Nested Design
6. Factorial Design
 - Two Factorial
 - Two Blocks
 - Four Blocks
 - General Factorial
7. Two-Level Fractional Factorial Designs
8. Hierarchical Design

Variants of the above procedures can be used to structure inputs to any experimental design to insure that the simulation requirements outlined in Section 2.2 are met.

Data will probably not come in the manner or order required for the metamodel. Therefore, to **gather data**, we must modify the simulation to add programs and routines to compute and collect required variables. We must insure that we modify the simulation only as necessary to capture the data. Variable names for the data structure should come from an external file that is built with the data file (See Figure 10.2.5).

Figure 10.4.19 shows how we will use prior knowledge of the metamodel representation and order or prior knowledge from the feature vector to select an ID routine (the routines here were covered above). From the data acquired from the simulation, determine and compute metamodel inputs.

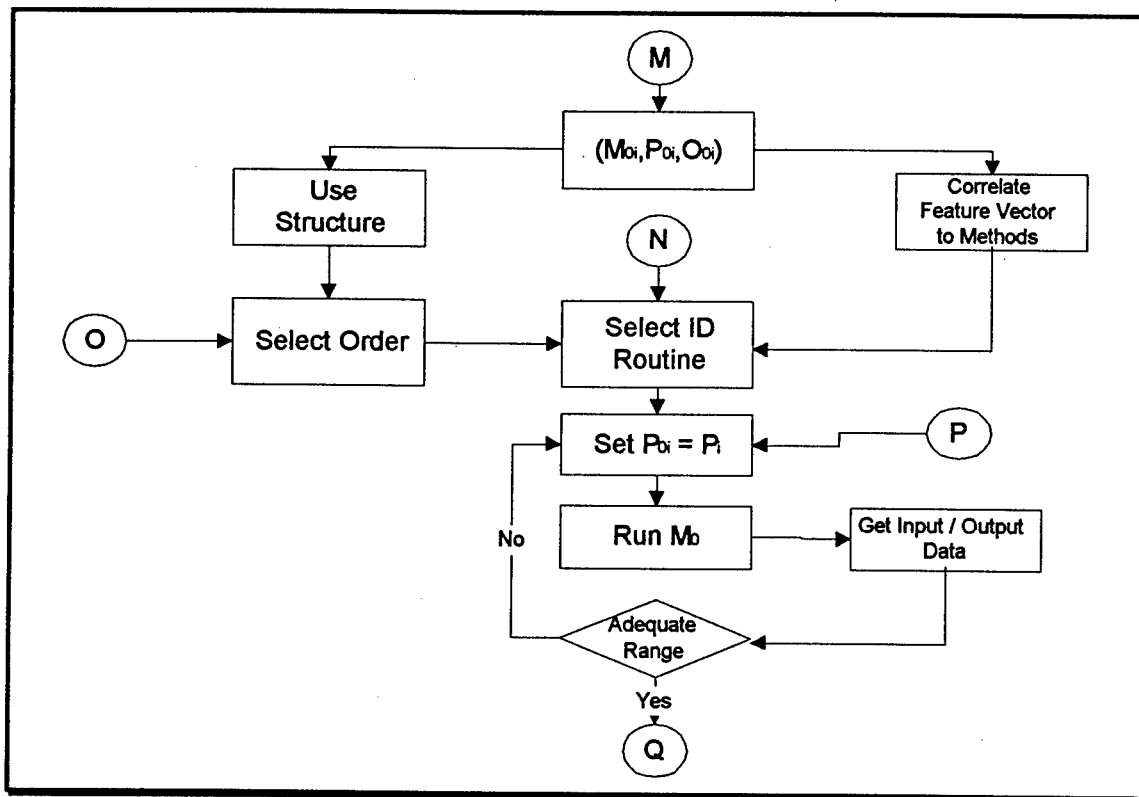


Figure 10.4.19 Data Acquisition Sequence.

Once determined, analyze the data that will be the input to the metamodel (see Figure 10.4.20). If calculations must be made for final or intermediate variables, extract the code directly from the simulation. Review the data provided by the simulation (verify the metamodel input). Use regression diagnostics (Chapter 9) to detect and remove co-linearity, since co-linearity reduces the frequency content of the identification input rendering the system rank deficient and removing the persistent excitation. Also identify influential observations and outliers.

If necessary, scale and center the metamodel inputs or detrend (remove the best straight-line fit). Select useful portions of the data. If required, filter the data to enhance

important frequency ranges. Clearly isolate (split) the data for each process and insure that the range of the input data is consistent with the purpose of the metamodel. If the output of the model is from $0+$ to ∞ , and the region of interest is 0 to $+5$. Then input data corresponding to the region of interest should be used for the model. If it is possible to split the model into one process that generates results in the area of interest, that is even better. Otherwise limit the input data to the area of interest.

A quasi-stationary infinite data set is called "informative" if it allows us to distinguish between different models in a set. Compute the spectrum matrix for $z(t) = [u(t)y(t)]^T$ and verify that it is strictly positive definite for all ω . Compare the information content (data statistics) versus degrees of freedom (number of parameters). Too many parameters will result in identifiability problems, too few parameters will result in poor performance

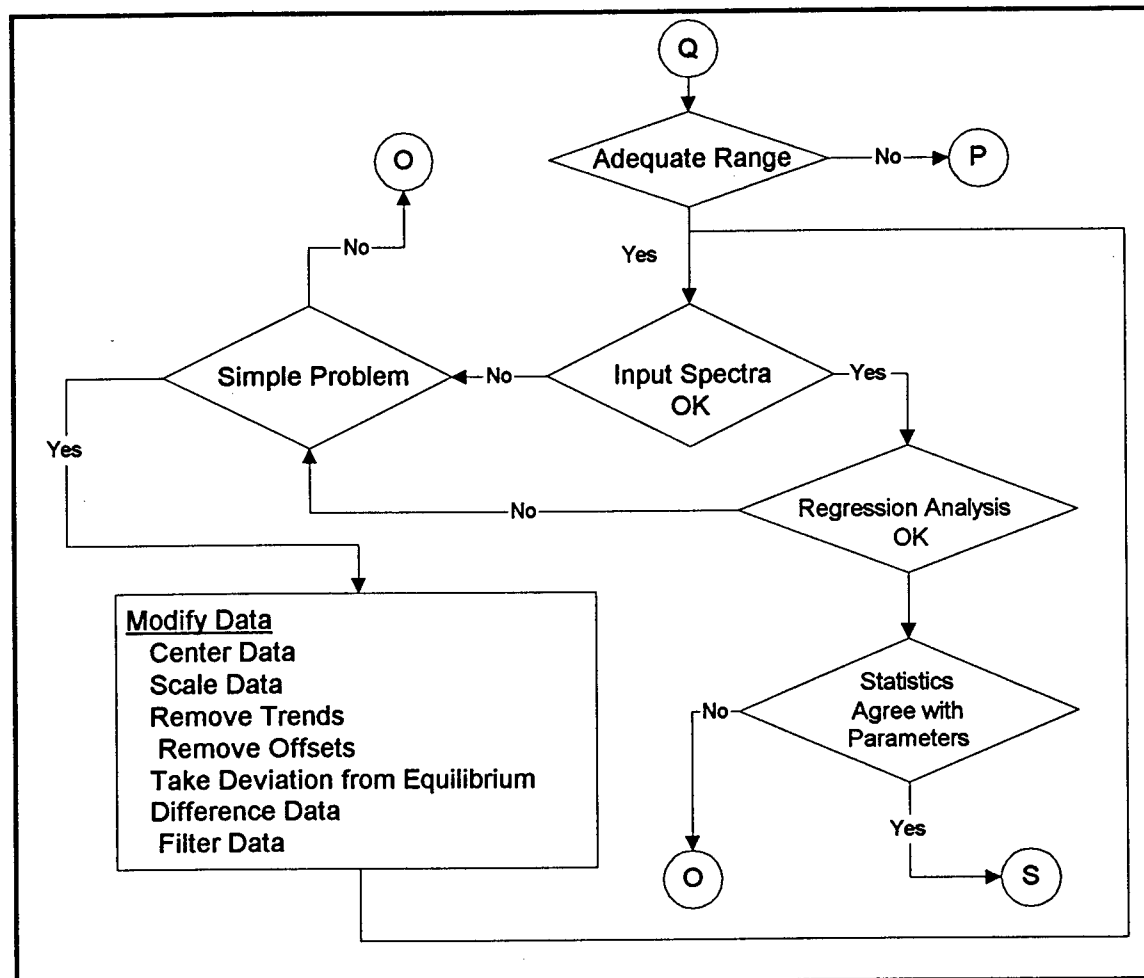


Figure 10.4.20 Metamodel Input Data Analysis Sequence.

4.4. Fit the Metamodel

At this point we have the metamodel representation (including the criterion of fit and identification method), the input data, and the validity measures. The next step is to parameterize the metamodel using an appropriate numerical method. The procedure to accomplish this is shown in Figure 10.4.21.

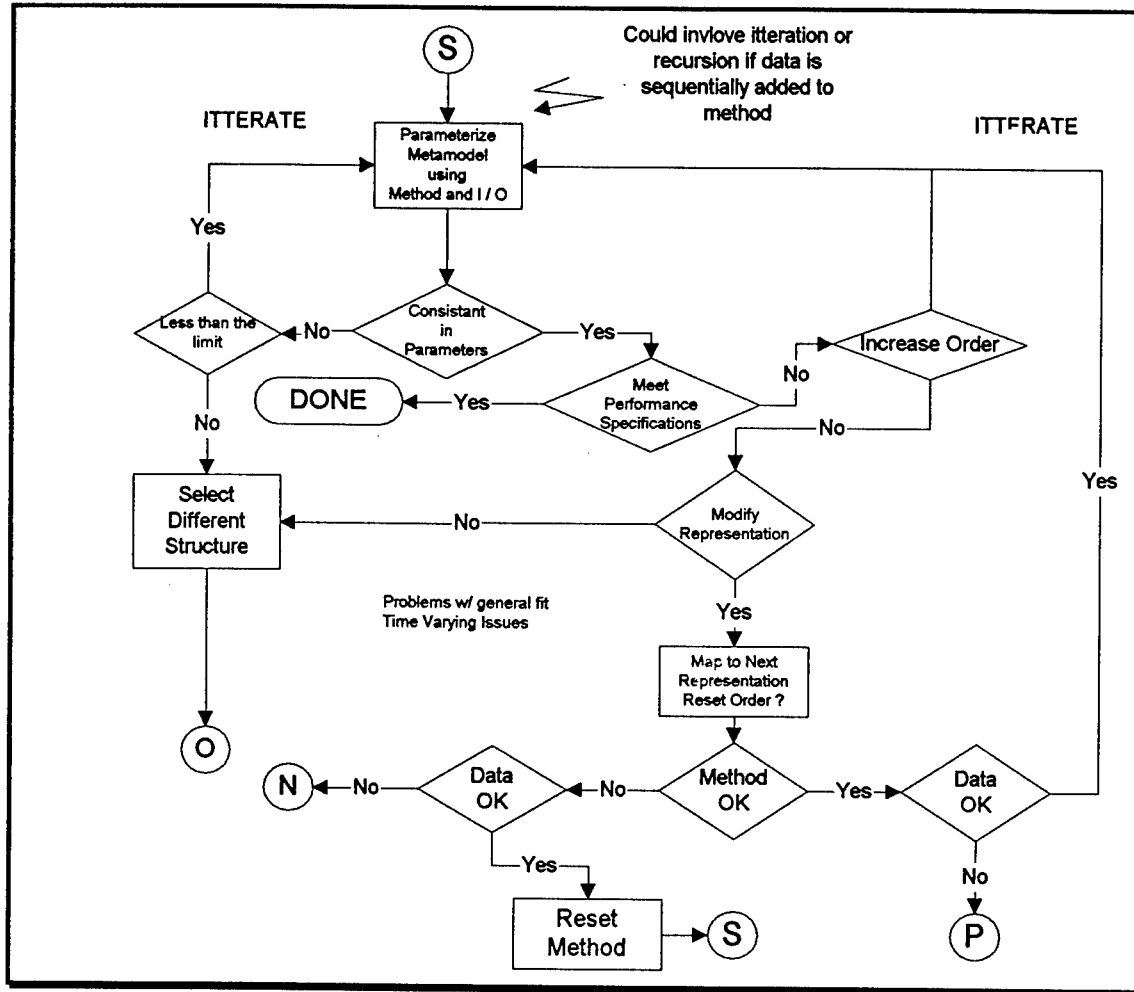


Figure 10.4.21 Generating the Metamodel.

4.5. Verify the Metamodel

Here we assess the validity of the metamodel. Table 8.3.1 lists the errors, their sources, and some of the causes or corrective measures. Validation must address all of the sources of error and must validate both the input and model assumptions. Chapter 8 presented the information from the perspective of information theory. Here we discuss validity from the analysts perspective in terms of metamodel performance, uncertainty, and adequacy.

4.5.1. Performance Analysis

In performance analysis, we insure that the data meet the assumptions of the identification method and that there are no problems with identifiability. We first address the local measures of validity, the consistency (bias) and variance of the solution. Each of the techniques will produce an internal measure of the variance of the solution. This value should be compared to the Cramér-Rao lower bound. Use the estimate of the bias and variance to compute a confidence ellipsoid. Compare the insensitivity to the Cramér-Rao lower bound.

A graphical comparison of the metamodel and simulations results can highlight errors or trends. We should determine output ranges based on the domain of the input variables, determine the sensitivity of the system performance on input parameters, and analyze the precision of the result.

4.5.2. Model Accuracy

If the assumptions were met, we now analyze the prediction errors to insure that they are within the error limits determined during problem definition. Measures that support this analysis are: graphic comparison maximum absolute error average absolute error; average absolute relative error; Akaike's information theoretic criterion (AIC) and Akaike's final prediction error (FPE).

4.5.3. Model Uncertainty

The performance of the identification method and the accuracy of the model was adequate. Now we should take a closer look at the variation in the parameters with hypothesis testing for significance.

4.5.4. Model Adequacy

The final set of validity measures are those that address the adequacy of the model in term of explaining the behavior observed in the data. This is accomplished by analysis of variance, residual analysis, goodness of fit and lack of fit testing, the squared coefficient of determination, correlation analysis, and spectral analysis.

5. PROCEDURAL NOTES

Assume that we have chosen a standard state space formulation for the model set.

For multi-output black box models it is easiest to try black box models. Start with ARX with a structure that is filled with parameters. Then consider those estimates that are the same magnitude as their standard deviations, and try orders and delays that automatically set them to zero. When a reasonable structure has been found try IV4 with it.

The problem with multivariable models is with the initial conditions. Therefore, fit individual matrices first, then fit the combination one at a time, then fit the combination with both free, and then finally fit the noise model. To determine which matrix should be determined first, do a CVA with the outputs only and then compare them to a input-output CVA to see where the greatest correlation is. If the input-output has the greatest correlation identify the B matrix first. Otherwise id the A matrix then the B matrix.

Selection of Model Order. The order of the process is evident in the autocorrelation of the output. For a perfect fit, the model order (sum of the order of individual variables and combinations) must equal the output correlation. CVA provides a measure of the variability of the data that gives the number and order of input variables.

Since the most general linear predictor routine is PEM, we discuss some procedures for multivariable systems. We are most interested in performance, and PEM is a compromise between fitting the transfer function and noise spectrum. We should concentrate on the identification of the transfer function by having a fixed noise model first.

If the signal-to-noise is not good, and it is important to have models that describe noise characteristics; try state space models in canonical form. These are equivalent to multivariable ARMAX models. Initial estimates can be set randomly. Canonical forms cannot handle input delays.

7. REFERENCES

1. L. Ljung, *System Identification: Theory for the User*, Prentice-Hall, New Jersey, 1987.
2. Press et. al., *Numerical Recipes*, Cambridge University Press, 1986.
3. M. Pait and S. Morse, "A Cyclic Switching Strategy for Parameter-Adaptive Control," *IEEE Transactions on Automatic Control*, Vol. 39, No. 6, June 1994.
4. Franklin, Powell, Workman, *Digital Control of Dynamic Systems*, 2nd Edition, Addison-Wesley Publishing Co., Reading MA, 1990.
5. D. Caughlin, "A Metamodeling Approach to Model Abstraction," *Proc. 1994 Fourth Annual IEEE Dual Use Technologies and Applications Conference*, May 1994.
6. D. Caughlin, "An Evaluation of Simulated Annealing for Modeling Air Combat Simulations," *Proc. 1994 Fourth Annual IEEE Dual Use Technologies and Applications Conference*, May 1994.
7. M. A. Zeimer, et. al., "Metamodel Procedures for Air Engagement Simulation Models," *IRAE Technical Report*, January 1993.
8. Sinha, Kusta, *Modeling and Identification of Dynamic Systems*, Van Nostrand, 1983.
9. J.P. Norton, *An Introduction to Identification*, Academic Press, San Diego, CA, 1986.
10. G. Milne, *State-Space Identification Tool for use with MATLAB*, The Math Works Inc., March 1988.
11. S.C. Agrawal, *Metamodeling, A Study in Approximations in Queuing Methods*, The MIT Press, Cambridge, MA, 1985.
12. Kendall, Stuart, and Ord, *Kendall's Advanced Theory of Statistics*, Volume 1, Distribution Theory, Oxford University Press, 1987.
13. G. Box, W. Hunter, J. Stuart, *Statistics for Experimenters*, John Wiley & Sons, New York, 1978
14. H. Cramer, *Mathematical Methods of Statistics*, Princeton University Press, 1963

CHAPTER 11

ADDITIONAL RESEARCH ISSUES

1. CHAPTER OUTLINE

1. Chapter Outline	11-1
2. Introduction.....	11-1
3. Theoretical Aspects	11-2
3.1. Selection of the Model Set and Order.....	11-2
3.2. Consistency of the Solution and Parameter Confidence Intervals.....	11-2
3.3. Algorithm Optimization.....	11-2
3.4. Optimal Input Design	11-2
3.5. Canonical Variate Analysis (CVA)	11-2
3.6. Matrix Fraction Descriptions and Block Structured, Norm-Bounded Uncertainty	11-3
3.7. Information Content Versus Degrees of Freedom and Data Statistics.....	11-3
3.8. Applications.....	11-3
4. System Development	11-4
4.1. Introduction	11-4
4.2. Discussion.....	11-4
5. References.....	11-6

2. INTRODUCTION

There are two areas where additional research is warranted. The first area is the theoretical aspects of the technique. System identification, realization theory, and statistics form the basis of the approach developed by this research. These areas are a dynamic area of study. All of the recent developments in these fields could not be incorporated into this research effort. Known references that should be investigated are included in the references to this chapter.

The second area needing additional work is the development of a system that will build on the capability to develop dynamic metamodels of tactical combat simulations. This system should provide the analyst with the ability to use metamodeling for software reuse, large scale model integration, verification, and validation.

Each of these general areas will be covered separately.

3. THEORETICAL ASPECTS

3.1. Selection of the Model Set and Order

While considerable progress was made in the selection of the model set and order, time-varying systems were not directly addressed. Recent work on time-varying parameter estimation should be evaluated [1].

We have relied on Canonical Variate Analysis to determine the model order. The order of the process, however, should be evident in the autocorrelation of the output. For a perfect fit, the model order (sum of the order of individual variables and combinations) should equal the output correlation. Additional work in this area may be beneficial.

3.2. Consistency of the Solution and Parameter Confidence Intervals

Additional work in solution consistency and in the calculation of the confidence ellipsoid would help in the validation process [2,3].

3.3. Algorithm Optimization

There is considerable effort in the optimization of the algorithms that was not included in this research. Topics include more efficient algorithms [4], bounded disturbances [5], adaptive control [6,7,8], reasonableness checks through fault detection and identification procedures [9], and new algorithms for a Markov chains [10],

Some times 2 to 3 hours are required to compute a parameter estimate. Parallel algorithms could be used to address this issue.

There are game theoretic approaches to identification. Algorithms for optimizing a set of mathematical objectives that have to be optimized simultaneously require: (1) a precise concept of optimality; and (2) decision making modules that allow for combination of criteria [11]. The integration can be based on non-cooperative games where no coalitions can be formed among the players. A Nash equilibrium can be used which is the optimal point for each module assuming that each of the other players is operating at its Nash equilibrium.

3.4. Optimal Input Design

Additional work could be done in experimental design [12,13,14,15].

3.5. Canonical Variate Analysis (CVA)

CVA shows promise as the most robust nonlinear identification method. Unfortunately, the technique is optimized to address time series without inputs and the variables grow explosively with multiple inputs. Research on the optimal structure of past and future vectors when inputs are considered is needed to address dimensionality issues.

3.6. Matrix Fraction Descriptions and Block Structured, Norm-Bounded Uncertainty

The most common error model used for identification assumes that all of the error enters the system as additive noise [16]. In this research, identification of multi-input multi-output metamodels was limited to the state space form. Matrix Fraction Descriptions (MFD) has formed the basis of H_∞ control theory and allows consideration of on a block structured, norm-bounded uncertainty that enters the model in a linear fractional manner.

Inclusion of this research into an metamodeling procedure should be a major topic for future research [17,18,19].

3.7. Information Content Versus Degrees of Freedom and Data Statistics

Additional work could be devoted to the analysis of the data, before the metamodel is generated to determine that the data meets the assumptions of the technique.

The experimental design must provide input-output sequences that correctly represent the system structure. Unfortunately, determination that the data contains the correct representation of the system structure cannot be made before the generation of the metamodel. Only when we have the metamodel can we validate the data by identifying the probability of the data set given the parameters as the likelihood of the parameters given the data

The use of information theory to determine how much variation can be explained by the selected number of parameters would help in the selection of the model order and set. The question of missing data should also be addressed [20].

Our current requirements for metamodeling of Discrete Event Simulations is restrictive. Additional research into premature termination, where autonomous behavior does not reach steady state, may help address experimental design questions.

3.8. Applications

Experience with additional situations would be informative. Metamodeling of a single realization of a simulation, multiple realizations with different initial conditions, a Monte-Carlo ensemble of the same initial conditions should all be addressed.

Application of reduced order metamodeling to the Verification, Validation, and Accreditation Issue shows great promise. This VV&A activity should include the use of metamodeling to determine the validity of training systems by comparing their results to the high fidelity M&S they are supposed to replicate.

SYSTEM DEVELOPMENT

Introduction

Model abstraction using metamodeling has demonstrated the capability to facilitate software reuse, large scale model integration, verification, and validation. This capability results from a new approach supported by a taxonomy of metamodeling problems, solution structures, and metamodeling methods. While these new methods work well, additional research is needed to build a robust system that will support the subject matter expert. This system should assist the analyst who is not familiar with model abstraction techniques but needs to reuse a piece of code, integrate different models, or verify a new version of a simulation.

Goal. Build on the capability to develop dynamic metamodels of tactical combat simulations. Provide the analyst with the ability to use metamodeling for software reuse, large scale model integration, verification, and validation

Objective. The objective of this portion of the followon research is to:

1. Build on existing procedures and algorithms to develop a metamodeling system.
2. Use this system to metamodel a broad class of simulations and investigate knowledge-base support required to make the capability readily accessible to the analyst.
3. Investigate the use of metamodeling in the Distributed Interactive Simulation (DIS) environment. Provide additional metamodeling support as required.

Discussion

The development of a metamodel still requires a thorough understanding of model abstraction, reduced order modeling, and system identification. In addition, even with the most robust procedures it is possible that the desired data generated by a simulation will not meet the assumptions or numerical requirements of the procedure.

Consequently, the widespread use of metamodeling as a method of model abstraction will require a fairly automated support system to assist the analyst. Development of this system is the primary objective of this type of research.

Metamodeling is a powerful tool. However, metamodeling is still in the research stage. As a tool, it will not be accepted by the simulation community until it demonstrates the ability to address current problems. Demonstration of metamodeling as a cost effective analytical and simulation tool is required.

Proposed Objectives

Objective 1. Build on existing robust procedures and algorithms to develop a metamodeling system.

Subobjective 1.1. Develop an environment for development of metamodels.

Subobjective 1.2. Provide the capability to analyze the source code, generate and run the simulation, and gather data.

Subobjective 1.3. Integrate metamodeling routines and procedures to generate and verify the metamodel.

Objective 2. Use this system to metamodel a cross section of simulations. Investigate the knowledge-base support required.

Subobjective 2.1. Metamodel simulations of interest to Rome Laboratory.

Subobjective 2.2. Based on the results of metamodeling, determine the type and amount of support required for the general user.

Subobjective 2.3. To the extent required, develop an expert assisted graphical interface, coupled with a knowledge-based expert system, to provide the user an easy and efficient way to accurately setup problems.

Objective 3. Investigate the use of metamodeling in the Distributed Interactive Simulation (DIS) environment.

Subobjective 3.1. Investigate the ability of metamodeling to support DIS.

Subobjective 3.1. Investigate the use of model abstraction to support the verification, validation, and accreditation process.

5. REFERENCES

1. J. E. Bobrow and W. Murray, "An Algorithm for RLS Identification of Parameters that Vary Quickly with Time," *IEEE Trans. on Automat. Contr.*, vol 38, no. 2, pp 351-354, February 1993.
2. T. E. Duncan, P. Mandl, and B. Pasik-Duncan, "Control Theory Methods for Consistency in Some Least Squares Identification Problems," *IEEE Trans. on Automat. Contr.*, vol 38, no. 8, pp 1289-1292, August 1993.
3. M-F. Cheung, S. Yurkovich, and K. M. Passino, "An Optimal Volume Ellipsoid Algorithm for Parameter Set Estimation," *IEEE Trans. on Automat. Contr.*, vol 38, no. 8, pp 1292-1296, August 1993.
5. D. N. C. Tse, M. A. Dahleh and J. N. Tsitsiklis, "Optimal Asymptotic Identification Under Bounded Disturbances," *IEEE Trans. on Automat. Contr.*, vol 38, no. 8, pp 1176-1189, August 1993.
4. D. Burshtein, "An Efficient Algorithm for Calculating the Likelihood and Likelihood Gradient of ARMA Models," *IEEE Trans. on Automat. Contr.*, vol 38, no. 2, pp 336-340, February 1993.
6. R. Ortega, "On Morse's New Adaptive Controller: Parameter Convergence and Transient Performance," *IEEE Trans. on Automat. Contr.*, vol 38, no. 8, pp 1191-1202, August 1993.
7. J. Sun, "A Modified Model Reference Adaptive Control Scheme for Improved Transient Performance," *IEEE Trans. on Automat. Contr.*, vol 38, no. 8, pp 1255-1259, August 1993.
8. P. S. Lewis, "QR-based Algorithms for Multichannel adaptive Least Squares Lattice Filters," *IEEE Trans. Acoust., Speech, Signal Proc.*, vol 38, no. 3, pp 412-431, 1990.
9. M. Saif and Y. Guan, "A New Approach to Robust Fault Detection and Identification," *IEEE Trans. on Aero. and Elec. Sys.*, vol 29, no. 3, pp 685-695, July 1993.
10. G. Shanthikumar and R. G. Sergeant, "An Algorithmic Approach for a Special Class of Markov Chains," *Ops. Res. Letters*, vol 3, no. 2, pp 81-83, June 1984.
11. H.I. Bozma and J.S. Duncan, "A Game-Theoretic Approach to Integration of Modules," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 16, No. 11, November 1994.
12. G. Goodwin, "Experiment Design for System Identification," *Encyclopedia of Systems and Control*, (M. Singh, ed.), Pergamon Press, Oxford, 1987.

13. G. Goodwin and R. Payne, *Dynamic System Identification: Experiment Design and Data Analysis*, Academic Press, New York, 1977.
14. R. Mehra, "Choice of Input Signals," *Trends and Progress in Systems Identification*, (P. Eykhoff, ed.), Pergamon Press, Elmsford, New York, 1981.
15. M. Zarrop, "Optimal Experimental Design for Dynamic System Identification," Springer-Verlag, New York, 1979.
16. R.S. Smith and J. Doyle, "Model Validation: A Connection Between Robust Control and Identification, *IEEE Transactions on Automatic Control*, Vol. 37, No. 7, July 1992.
17. A. J. Helmicki, C. A. Jacobson, and C. N. Nett, "Identification in H_∞ : A Robustly Convergent, Nonlinear Algorithm," *Proc. 1990 Amer. Contr. Conf.*, pp 386-391, 1990.
18. A. J. Helmicki, C. A. Jacobson, and C. N. Nett, "Fundamentals of Control-Oriented System Identification and their Application for Identification in H_∞ ," *Proc. 1991 Amer. Contr. Conf., San Diego*, pp 89-99, 1991.
19. A. J. Helmicki, C. A. Jacobson, and C. N. Nett, "Least Squares Methods for H_∞ Control-Oriented System Identification," *IEEE Trans. on Automat. Contr.*, vol 38, no. 5, pp 819-826, May 1993.
20. A. J. Isaksson, "Identification of ARX-Models Subject to Missing Data," *IEEE Trans. on Automat. Contr.*, vol 38, no. 5, pp 813-819, May 1993.

CHAPTER 12

SUMMARY OF RESULTS

1. CHAPTER OUTLINE

1. Chapter Outline	12-1
2. Research Summary	12-1
2.1. Statement of the Problem	12-1
2.2. Metamodeling	12-1
2.3. Objectives	12-1
3. Results and Discussion.....	12-6

2. RESEARCH SUMMARY

2.1. Statement of the Problem

Tactical simulation models used by the Department of Defense to assess the capabilities of combat systems and tactics are highly complex. It is often difficult to determine the relationship of individual factors to the performance of the modeled process. Consequently, it is not easy to use the results of the model in another simulation or couple multiple models to investigate a larger issue. The result is a proliferation of point-designed models and simulations, expensive upgrade and maintenance, and the inability to efficiently answer many of the more difficult questions raised by the acquisition and operational communities. Recently, a technique called metamodeling has generated interest for its ability to facilitate this type of assessment.

2.2. Metamodeling

A metamodel is a mathematical approximation of the system relationships defined by a high fidelity model or simulation. As an approximation, it is a projection of the model onto a subspace defined by new constraints or regions of interest. Selection of the parameters used for the projection (the construction of a metamodel) involves: *a priori* knowledge, the data, a set of metamodel structures, and rules to determine the best model to realize the data. Metamodeling is an innovative technique on the verge of providing the Air Force significant increases in capability. This technique will impact a broad range of Air Force activities from the development of combat decision support systems to the integration of complex large scale simulations.

2.3. Objectives

The objectives of this research were to:

1. Define classes of Air Force metamodeling problems based on the simulations and *a priori* knowledge (metamodel use). Determine criteria for clustering metamodeling problems. Apply these criteria to selected simulations.

2. Categorize the set of available metamodel structures and determine criteria for application to Air Force metamodeling problems. Demonstrate use of these criteria.

We defined a metamodeling problem as the direct sum of the model (simulation) and metamodel requirements. This means that the same simulation could be part of two different metamodeling problems if the requirements were different. Or conversely, the same set of requirements applied to two different (nonsimilar) simulations also leads to two different metamodeling problems.

There are many techniques available to address an Air Force metamodeling problem. The issue is the proper use of systems theory and experimental design to arrive at the "best" metamodel set that solves a particular problem. A solid connection between the problem (prior knowledge) and solution technique was needed. Objective 1 provided the requirements background to address this connection between the problem and the structure. Objective 2 addressed the primary issues posed in this research. This part of the research concentrated on steps necessary to generate the metamodel. The result was be a cluster of metamodel problems and a procedure for defining the metamodel set.

Objective 1. The first objective of this research focused on the steps that provide the prior knowledge required to develop a metamodel:

1. Determine the purpose of the metamodel
2. Identify the response
3. Identify important response characteristics
4. Identify input factors
5. Identify important input characteristics
6. Specify the experimental region
7. Select validity measures
8. Specify required validity

The connection between prior knowledge and the metamodeling technique began with an analysis of the types of problems facing the Air Force analyst and engineer. In line with the definition of a metamodeling problem, this analysis began with the purpose of the metamodel and then addressed the characteristics of the simulations.

Definition of the purpose began with the identification of the user as either from the operational or acquisition community. Then, metamodels used for acquisition and operational purposes were grouped by objective as either analytical metamodels or simulation metamodels. An analytical metamodel is used for analysis. In this case, the metamodel becomes an independent structure that is used to understand and extract information from the model. A simulation metamodel is used to support hierarchical simulation and model reuse. Consequently, a simulation metamodel is used in conjunction with (coupled to) other simulations or simulation elements to answer larger questions that are not supported within the structure of the modeled simulation. For each of these groups, the potential scope and use of the metamodel was defined.

Characteristics of the simulations were grouped as internal and external. Internal characteristics focused on the basis of the simulation (physical or event driven), a description of the internal process with respect to complexity and coupling, and the system structure with which the input, output, and system was defined. External characteristics were based on the SIMTAX definitions of the Military Operational Research Society and included purpose, qualities, construction, time processing, treatment of randomness, and sidedness.

One hundred sixty two combat simulations were selected from the *Catalog of Wargaming and Military Simulation Models, 11th Edition*, compiled by the Force Structure, Resource, and Assignment Directorate (J-8). Using the simulations selected from the catalog, a binary feature space of dimension 125 was developed. A metric space using this feature vector was defined and the density of the metamodeling problems in the feature space was analyzed. Using the difference between feature vectors, clusters of simulations were determined, and a characteristic vector was defined for each subcategory entered into the database. This vector was used as the centroid of a cluster for the category and the distance from each simulation to all of the characteristic vectors (one defined for each category) was determined. From the analysis of these clusters and the distribution of simulations, we found that there is a structure to the selected simulations and that classes of metamodeling problems could be defined by these clusters.

This result successfully answered Objective 1.

Objective 2. Research for the second objective addressed the steps that define and determine the metamodel (Steps 9 through 13):

9. Postulate a metamodel based on:
 - Input - Output response characteristics
 - Experimental region dimensions
 - Required validity
10. Select an experimental design
11. Obtain data
12. Fit the metamodel
13. Assess the validity of the model

Metamodeling decisions associated with this objective are complex, interrelated, and not supported by a unifying theory or procedure. We separated the decisions for this objective into selection of the metamodel set, validity measures, and experimental design.

The theoretical background used for the framework of this research was significantly different from the usual approaches followed by either the operations research (analysis) or engineering communities. This new approach allowed us to develop a new taxonomy that classified available metamodel sets and methods of generating the metamodel in a manner that could directly support the above metamodeling decisions.

Given that multiple model sets are available, the model structure defines the allowable behaviors of the model that are allowed. The metamodeling structure included the system description that defined the representation and provided both predictor and probabilistic

models. Specifically the system description covered the system description, class, and metamodel structure.

For the system description, we assume that the system parameters are lumped (as opposed to distributed). We now classify the systems as either static or dynamic with a defined algebraic structure (linear or nonlinear) and treatment of randomness (deterministic or stochastic). Dynamic systems also are grouped by the propagation of time (continuous, discrete, or continuous-discrete).

For each system description we include the class of structure as: Single-Input-Single-Output (SISO); Multiple-Input-Single-Output (MISO); and Multiple-Input-Multiple-Output (MIMO). Both time invariant and time varying versions are discussed.

There are two general metamodel structures, predictor models and probabilistic models. A predictor model only defines the predictor equation(s). Predictor models are models that specify the elements of the transfer function in terms of some parameter. The models generated from these structures are deterministic in nature. A probabilistic model accommodates the fact that many systems are subject to known disturbances that are not (or cannot be) completely categorized. Probabilistic models supplement the parametric description with a description of the density function (or moments) of the noise (disturbance) that acts on the system. The discussion on probabilistic models also included the most realistic description: a nonlinear Ito stochastic model.

These two general structures can be represented by three forms. They can be expressed as a polynomial, a matrix fraction, or in a state space form.

The methods to generate the metamodel were classified by the form of the identifier and the criterion of fit. Within these selections, all of the following approaches were discussed in detail: prediction error methods, correlation approaches, maximum likelihood approaches, optimization, and approximation techniques. Again, this is a new taxonomy. This new classification simplifies the decision process by placing hundreds of specific techniques in one of five categories.

Having classified metamodel structures and methods of generating the metamodel, we then reviewed methods that assist in the determination of which model structure and order to select. We investigated issues affecting the selection of the structure and order, equivalent realizations, canonical forms, and the impact of minimal realizations on observability and identifiability. Since the development of a metamodel is often an iterative process, we provided general model order determination techniques suitable for a first attempt as well techniques that were applicable to particular methods. Techniques to determine the model order were then augmented by method of reducing the order of an identified model.

The research then focused on methods to assess the validity of the metamodel. We covered measures of local as well as global validity. Local validity measures concentrate on internal measures. There are two types of validity measures considered here. The first,

local validity, are measures of the properties of the parameters themselves. These consist of bias, variance, consistency, and efficiency. The second type of internal measure are properties of the identification method. These properties are characteristic of the criterion and identification method that was used to parameterize the model. Given that the error criterion was minimum mean square error, what was the mean square error? How does this error compare to the theoretically obtainable value?

While the local validity measures concentrate on internal measures of the model validity, the global measures are more focused on the ability of the model to represent the system. Again, there are two types of global validity measures. The first type is with respect to the general information content in the data. Does the model extract the maximum amount of information from the data? The second type of measure attempts to measure the validity by computing the accuracy of the model output. Each of the above validity measures is investigated and discussed.

The last topic we investigated before addressing specific procedures to metamodel combat simulations concerned the experimental design. The design of an experiment includes which variables to measure and when to measure them and which variables to manipulate and how to manipulate them. Experimental design structures the change to the input variables so that we may observe and identify the reasons for changes in the output response.

An analyst (or statistician) will spend significant time deciding how to draw a sample from the general population so that the data will conform to certain assumptions and allow valid statistical inference. Control system engineers take a different tact. They are usually trying to identify a model for a piece of equipment and will concentrate on insuring a persistently exciting input signal so that all of the system modes will be excited. We combined both elements of experimental design in a discussion of the principles and the impact of choices for both design approaches.

The final effort was to provide the connection between the metamodel problem and the solution technique. During the research it appeared that the possibility existed to develop a more robust identification procedure. The new metamodeling approach, combined with a more robust procedure, changed the nature of the connection between the metamodel problem and solution technique. The approach did a better job of connecting the solution to the problem and a more robust identification procedure required fewer metamodeling solution techniques.

Consequently, we concentrated on developing a more robust capability for multi-input-multi-output dynamic metamodels that would not be as fragile as previous techniques. Therefore, we developed the taxonomy of metamodeling problems, but did not build the knowledge base to connect the problem and the solution. Instead, The connection between the problem and the solution was made through the procedures developed to match the model set to the problem definition.

These procedures successfully answered Objective 2.

3. RESULTS AND DISCUSSION

RESULT: Developed a dynamic metamodeling procedure based on a system theoretic framework (Chapter 2, Section 4.3, pg. 2-8; Chapter 3, Section 2.3, pg. 3-3).

DISCUSSION: Expanded the use of metamodels from static models that mapped a set of inputs to the observed output to the identification of the underlying processes that define the system that generated the data. Therefore, we are trying to identify the underlying dynamical system. We stress dynamical systems because they exhibit memory and can model phenomena where the past influences the future.

RESULT: Defined metamodeling constraints. (Chapter 2, Section 4.5, pg. 2-9; Chapter 3, Section 5.2, pg. 3-13; Chapter 9, Section 3.3, pg. 9-11).

DISCUSSION: During the inverse modeling process, it is imperative to model only one system. Also, the metamodeling process assumes that the system or process to be modeled is complete. Therefore, it is not possible to metamodel part of a process.

If an attempt is made to model two systems, behaviors associated with both processes will be aliased preventing the identification of either. Usually a numerical problem will surface. Using the PEM, the rank deficiency in the uncoupled equations caused numerical difficulties. In MMLE, calculation of the Jacobians will be reduced rank because all of the outputs are not functions of all of the variables.

In the usual case of inverse modeling, the system is not known. However, in our case, the dynamics of the simulation are available so that it is possible to determine the number of processes that are present in the interconnected system.

Co-linearity also reduces the frequency content of the identification input rendering the system rank deficient and removing the persistent excitation.

RESULT: Clearly defined the metamodeling problem (Chapter 2, Section 6.1, pg. 2-11).

DISCUSSION: Defined the metamodeling problem as the direct sum of the metamodel requirements and the model (simulation). This allowed a solution by independently considering both elements of the direct sum -- the purpose of the metamodel and the simulation characteristics.

RESULT: Defined the conditions to metamodel discrete event simulations. Identified the identifiability issues (Chapter 3, Section 4.0, pg. 3-10; Chapter 3, Section 5.2, pg. 3-13).

DISCUSSION: Defined when a DES can be described by a difference equation. Discussed the termination requirements.

RESULT: Classified the general purposes of metamodels into analysis and hierarchical simulation (Chapter 4, Section 3.2, pg. 4-1).

DISCUSSION: First, a metamodel can be used for analysis. In this case, the metamodel becomes an independent structure that is used to understand and extract information from the model.

Furthermore, a metamodel can be used to support hierarchical simulation and model reuse. In this case, the metamodel is used in conjunction with (coupled to) other simulations or simulation elements to answer larger questions that are not supported within the structure of the single simulation.

The selections view of the metamodel purpose provided clear boundary conditions for follow-on selections.

RESULT: Clearly defined simulation characteristics (Chapter 4, Section 4.3, pg. 4-21).

DISCUSSION: Developed a new general description of the simulation or model. The external description was based on SIMTAX and was augmented by an internal description that provided further detail on the internal structure of components.

RESULT: Developed a binary feature space of simulations (for representation) (Chapter 4, Section 6.4, pg. 4-27).

DISCUSSION: A metric space using this feature vector was defined and the density of the metamodeling problems in the feature space was analyzed. Using the difference between feature vectors, clusters of simulations were determined, and a characteristic vector was defined for each subcategory entered into the database. This vector was used as the centroid of a cluster for the category and the distance from each simulation to all of the characteristic vectors (one defined for each category) was determined.

RESULT: Developed a new taxonomy of model representations (Chapter 5).

DISCUSSION: Reduced selection of the metamodel set to four sequential decisions based on *a priori* information. The first two decisions concern the **system description** and **class**. The next decision defines the **structure** of the metamodel, while the last selection provides the **identification methodology**. Each decision is bounded by preceding choices. This

process reduces the number of independent decisions required to develop a metamodel.

RESULT: Reduced the "myriad" of model structures to two: **predictor models** and **probabilistic models**. Developed a procedure to determine the model set (Chapter 5, Section 3.4, pg. 5-5).

DISCUSSION: We define two general model structures: **predictor models** and **probabilistic models**. A predictor model only defines the predictor equation(s). Predictor models are models that specify the elements of the transfer function in terms of some parameter set. Models generated from these structures are deterministic. Predictor models, however, do allow for both prediction and measurement error. And since the coefficients were generated through the minimization of an error function with assumed statistics, the coefficients will be random variables with a distribution. Since the estimates are functions of these random variables, this distribution can be used to compute error bounds of the estimate.

A probabilistic model accommodates the fact that many systems are subject to known disturbances that are not (or cannot be) completely categorized. The statistics of the noises and disturbances are included as random variables. Probabilistic models supplement the parametric description with a description of the density function (or moments) of the noise (disturbance) that acts on the system. The variables of the system being identified become functions of random variables. In these situations, different realizations of an experiment (simulation run) may not produce exactly the same results. Consequently, the output of a probabilistic model is both the conditional expected value and probability density functions (CPDF) of the variables.

RESULT: Defined the three forms of representation (Chapter 5, Section 4.2, pg. 5-10).

DISCUSSION: They can be expressed as a polynomial, a matrix fraction, or in a state space form.

RESULT: The classified methods to generate the metamodel by: the **form of the identifier** and the **criterion of fit** (Chapter 6, Section 2.3, pg. 6-8).

DISCUSSION: Parameter identification methods are used when the candidate model is to be defined by a set of parameters. The form of the identifier defines the "experimental setup" (Equation Error Method, Output Error Method, and Prediction Error Method) or the manner in which the estimates are generated and compared. From within these selections, all of the following approaches were discussed in detail: prediction error methods, correlation

approaches, maximum likelihood approaches, optimization, and approximation techniques.

By criterion of fit, we mean the function that is optimized to determine the parameter estimates. The criterion of fit establishes both the cost function and the method of its minimization. There are three: minimum mean square, maximum a posteriori (maximize the CPDF), and maximum likelihood (maximize the Joint PDF).

RESULT: Developed a full nonlinear identification technique in Canonical Variate Analysis (Chapter 6, Section 7.4, pg. 6-44).

DISCUSSION: Developed the ability to identify nonlinear time series. Identified a problem with dimensionality of multivariable CVA.

RESULT: Developed procedures that determined which inputs and outputs to use (Chapter 7 and Chapter 10).

DISCUSSION: Determined what data to use - defines the domain of the input. Determined how to structure this input. Used CVA to provide a measure of the variability of the data that gives the number and order of input variables. Provided methods to determine if the parameters are too many or too few.

RESULT: Every metamodel is an approximation. The input domain and output range where the metamodel is valid must be determined (Chapter 8).

DISCUSSION: Given a known system, every projection of that system into a subspace will reduce the information content of the observed behavior. The only exception to this is the situation where the kernel of the projection coincides with the null space of the behavior.

RESULT: Expanded validity measures and experimental design options (Chapter 8, Chapter 9).

DISCUSSION: Statistical experimental design and validation methods were augmented with control system design and validation procedures. Discussed the nature of the residuals.

RESULT: Restructured the metamodeling process as elements of a more general structure that directly coupled the *a priori* knowledge to the structure of the metamodel (Chapter 10, Section 4, pg. 10-11).

DISCUSSION: The revised process is a set of sequential decisions based on *a priori* information that reduces the number of independent decisions required to develop a metamodel. This process provides a direct method of sorting through the myriad of decisions necessary to develop a metamodel and is

supported by a set of computer capable routines that match the problem definition with the simulation characteristics.

This process is also supported by a new taxonomy of metamodel structures and methods to generate the metamodel and allows the separation of the metamodeling process into a few well defined steps. The first eight steps became the foundation for the **problem definition**, the remaining steps were grouped in an iterative scheme as the **metamodeling process**.

RESULT: Determined that the ability to metamodel components of simulations is a significant function of the structure of the code (Volume 2, Chapter 2; Volume 2, Chapter 4).

DISCUSSION: Code that relies on global (common) data structures where components of the data are calculated in several different modules may be difficult to metamodel without significant modifications.

RESULT: Developed a more robust metamodeling procedure that could be applied to a broader range of problems than existing techniques (Software reference manual).

DISCUSSION: During the research it appeared that the possibility existed to develop a more robust identification procedure. The new approach, combined with a more robust procedure, changed the importance of the connection between the metamodel problem and solution technique. The approach did a better job of connecting the solution to the problem. The more robust technique required fewer metamodeling solution techniques and would not be as fragile as previous procedures.

Therefore, we developed the taxonomy of metamodeling problems, but did not build the knowledge-base to connect the problem and the solution. The connection between the problem and the solution was made through the procedure as opposed to a relationship between the classes of problems and solution techniques.

CHAPTER 13

CONCLUSIONS AND RECOMMENDATIONS

1. CONCLUSIONS

This research met the objectives and demonstrated the viability of metamodeling dynamic systems.

In addition to clearly defining all aspects of the problem, this research had a number of significant successes. The research expanded on the metamodeling approach to directly couple the *a priori* knowledge to the structure of the metamodel. It also developed a taxonomy of simulations that defined classes of metamodeling problems which could be used to specify which metamodeling method was most appropriate. A more robust metamodeling procedure was developed that could be applied to a broader range of problems than existing techniques.

We have taken the standard metamodeling procedures and structured the process into a set of sequential decisions based on *a priori* information. This process reduces the number of independent decisions required to develop a metamodel.

This enhanced process is supported by a new taxonomy of metamodel structures and methods to generate the metamodel and allows the separation of the metamodeling process into a few well-defined steps.

2. RECOMMENDATIONS

There are two general recommendations that come out of this research.

1. The first recommendation is to build on the capability to develop dynamic metamodels of tactical combat simulations and implement the revised metamodeling process described in Chapter 10.
2. The second recommendation is to support development of the expert assisted prototype metamodeling system outlined in Chapter 11, "Additional Research Issues."

APPENDIX 1

SIMULATION IN THE AIR FORCE

1. CHAPTER OUTLINE

1. Chapter Outline.....	A1-1
2. Introduction.....	A1-1
3. Requirement	A1-1
4. Capability Provided.....	A1-2
5. Levels of Analysis	A1-2
5.1. Hierarchy of Models.....	A1-3
6. Use of Simulation in the Air Force	A1-3
7. Modeling and Simulation in the Acquisition Process.....	A1-5
7.1. Mission Area Analysis	A1-5
7.1.1. Capability Shortfalls.....	A1-6
7.1.2. Capability with New System or Operating Concept	A1-6
7.2. Concept Exploration Phase (Milestone 0)	A1-6
7.3. Demonstration and Validation phase (Milestone 1)	A1-7
7.4. Engineering and Manufacturing Development Phase (Milestone 2).....	A1-8
7.5. Production and Deployment Period (Milestone 3).....	A1-8
8. Modeling and Simulation in the Operational Environment	A1-9
9. References	A1-10

2. INTRODUCTION

Since the focus of the research is on the metamodeling of combat simulations, the purpose of the metamodel is constrained by the use of simulations in the Air Force. A summary of this use follows.

3. REQUIREMENT

The requirement for good modeling and simulation has been recognized for over 20 years. The Air Force-Wide Mission Area Analysis Model was a response to the Congressional Budget and Impoundment Control Act of 1974. The shortcomings of current processes were clearly recognized by the Electronic Combat Broad Area Review in 1988.

Simulation activities currently underway in the Air Force involve investigations and analyses at a variety of levels, embodying the ideas of the hierarchy of models that will be discussed below.

Solid systems engineering requires an understanding of the performance of the system in its intended environment. The outcome of combat, however, is not merely driven by the individual capabilities of the systems. Combat is a complex interaction of capability, information, strategy, and tactics. The Air Force has the requirement to maintain the ability to model many aspects of combat and provide decision makers with an indication of how well current systems compete with potential adversaries.

4. CAPABILITY PROVIDED

There are two primary capabilities provided by modeling and simulation (M&S). First, the results of simulation provide **quantitative information** to decision makers. They are able to analyze the "real world" environment and determine the impact of potential decisions. In addition to the impact of their decisions, M&S also allows the decision maker to "completely" characterize the process, resulting in better understanding of the issues.

In addition to providing information for decision makers, M&S structures the acquisition process. In particular, it provides an indication of critical technologies and capabilities that must be available for success, and outlines the test and evaluation (T&E) program necessary to support the development.

5. LEVELS OF ANALYSIS

In order to understand the use of simulations, the analyst must understand how the simulation fits into the overall "hierarchy of models". There are four levels in this hierarchy:

1. **Level I.** Analysis at this level primarily deals with individual systems or components. The objective is to understand the physical processes involved and develop the required capability. Once developed, the performance is quantified for use in higher level simulations [1]. The analysis at this level is usually limited to the effects of a single component accomplishing a specific task.
2. **Level II.** At this level, the evaluation focuses on the component being associated with a platform; e.g., a radar installed on an aircraft. The effectiveness of the installed system is then evaluated in the context of a specific task and is usually a one-on-one engagement analysis.
3. **Level III.** This level of analysis assesses the contribution of the platform along with the tactics or methods in a combat mission environment. This environment includes other aspects such as mutual support, command and control, required maneuvers, and a defined order-of-battle.
4. **Level IV.** This encompasses all the activity associated with operations in the context of a joint Air Force/Army/Navy campaign against an enemy combined arms force, towards evaluating the contribution of, for example, electronic combat support in such a campaign.

5.1. Hierarchy of Models

Model types used in the acquisition process can be arranged in a hierarchy associated with the level of analysis. This hierarchy has the most detailed physical process models at the base and the most aggregate force-on-force models at the top. In general, models of physical processes are very faithful representations of reality, but as we go up the hierarchy to one-on-one models, many-on-many models, and finally to force-on-force models, fidelity tends to decline. Physical process and one-on-one models are used predominately for test and evaluation of systems, whereas many-on-many and force-on-force models are predominately used in making policy or management decisions. The relationship between the level of analysis and focus of the simulation is shown in Table A.1.5.1.

NOTE: The minutes of the of the HTI Modeling and Simulation (M&S) Working Group Kickoff Meeting on 10 September 1992, depict five levels of simulations. Level III, "Many-On-Many", has been split into two groups. The first group is "One-On-Many" or "Few-On-Few", the second group is "Many-On-Many". This separation better highlights the complexity of the simulations, but does not materially change the concept of the hierarchy.

Table A1.5.1 Hierarchy of Models.

LEVEL	SIMULATION PURPOSE	SIMULATION FOCUS
Level I	Engineering analysis	Physical process
Level II	Weapon system capability	One-On-One
Level III	Combat capability	Many-on-Many
Level IV	Campaign results	Force-on-Force

6. USE OF SIMULATION IN THE AIR FORCE

One of the reasons that M&S is such a complex subject is the multiple purposes it can be used for. Whatever its use, the simulation should focus on major impacts. To be most effective for the Air Force, the simulation should focus on the issues that have the most impact on operational capability. Initially the focus of the simulation effort will be to identify deficiencies, evaluate different options to correct the deficiency, and secure approval of **concept development**. The next major area for simulation is the accomplishment and refinement of **system design**. In this phase the major objective is to avoid development failures. An ancillary benefit to this objective will be the definition of

the T&E program. From this point forward the simulation can be used to develop and **modify the concept of operations**. With the focus of simulation in mind, some of its specific uses are highlighted:

1. Evaluate specific techniques.
2. Decision tool.
3. Collateral impacts of changes.
4. Tradeoff analysis.

There are two general areas that metamodels can support. These are acquisition (including the total integrated weapon system support) and operations (including the logistic as well as the employment of the system):

1. Acquisition is the process by which weapon systems are acquired and supported. Major acquisition programs are managed from a structure that is separate from the operational chains of command. In acquisition, a metamodel will directly support one of the phases of the acquisition process (which includes a production and operations period).
2. Operations is the deployment and employment of weapon systems and personnel. In operational use, a metamodel is used to exploit the military utility of an existing system by defining or improving operational procedures, tactics, or strategy.

Figure A1.6.1 depicts the lifecycle process of an Air Force System [2].

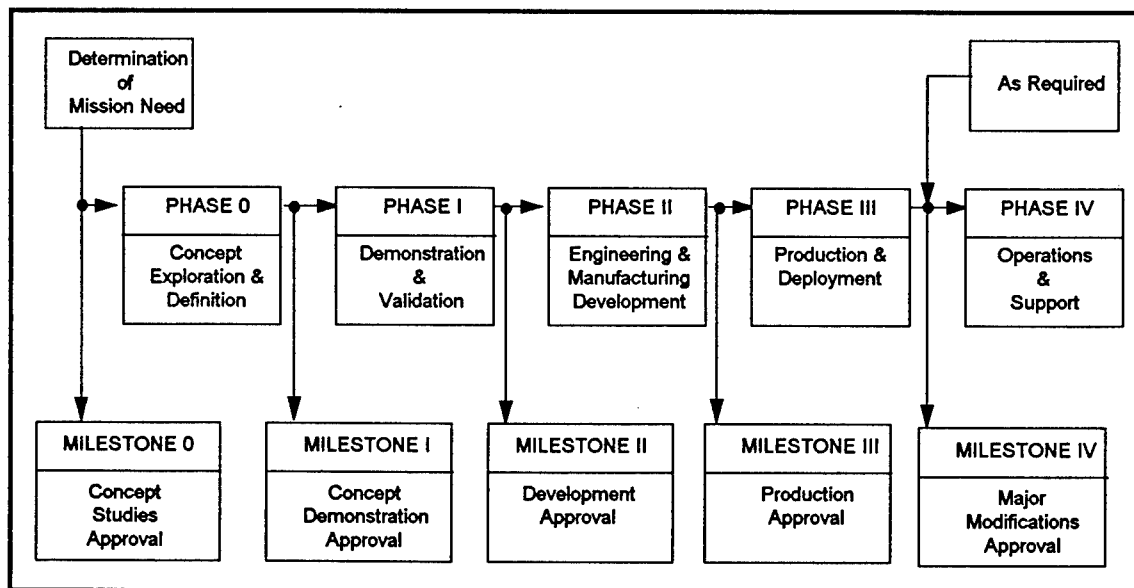


Figure A1.6.1. Air Force System Lifecycle.

7. MODELING AND SIMULATION IN THE ACQUISITION PROCESS

This section discusses how M&S is used to support the Air Force acquisition process. Several examples of the use of specific simulations and their impact on a particular program are provided.

The DoD acquisition process is divided into several formal phases from concept exploration through demonstration/validation, engineering and manufacturing development (including development testing), production and deployment (including operational testing), to actual operations and support. For a given weapon system, these formal phases are preceded by an Air Force-unique period of study termed Mission Area Analysis.

Quite naturally, each of these phases raise different analysis issues and consequently require different modeling and simulation capabilities. For example, the Mission Area Analysis period deals with the questions: (1) Do we have capability shortfalls?, and (2) Could our capability be improved with a new system or operating concept? In the Concept Exploration Phase of acquiring a new system, analysis focuses on potential system performance, impacts on force capability or effectiveness, and affordability. In the Demonstration/Validation phase, we look at performance parameter balance and tradeoffs, the achievability and affordability of theoretical system performance, and whether the system will make a difference in overall force effectiveness. And of course, during Engineering and Manufacturing Development, which includes development testing, the main analysis issue is whether the new system meets specifications established as a result of analyses in the preceding phases. Finally, in the Production and Operations Phase, which includes operational testing, when the user is actually employing the system, the paramount issue is "Does the system meet the user's needs?"

7.1. Mission Area Analysis

In the Mission Area Analysis stage that precedes the formal phases of the acquisition process, the main thrust is to **identify shortfalls** in Air Force capability to meet the projected threat. Mission Area Analysis is also used to estimate the potential impact of **emerging technologies** and **alternative tactics** and operational concepts on weapon system performance and overall decision capability. Ultimately, the analysis is used to **predict force effectiveness** in future war scenarios.

The actual Mission Area Analysis model is a hierarchical model that uses the analytical hierarchy process (AHP) methodology [3]. The model incorporates broad Air Force mission areas, war scenarios, geographic theaters, major commands, and specific mission tasks and objectives. The model is exercised annually using alternative weapon system mixes and force structures being considered by the Air Staff for inclusion in the Air Force Program Objective Memorandum (POM) to the Office of the Secretary of Defense. The

resulting model-projected capability is then fed back to the Air Staff as an input to the corporate decision making process.

Mission Area Analysis has a positive and useful impact on the Air Force programming system by assessing weapon system and force structure alternatives in terms of mission capability. These assessments are often used as one of several inputs to the corporate process which decides how many of what weapon systems the Air Force will formally propose to acquire.

In the time-compressed environment of Air Force Program Objective Memorandum deliberations, which is part of the DoD Planning Programming and Budgeting System, supporting analyses, if they are to be of any use at all, must be both credible and timely.

7.1.1. Capability Shortfalls.

What are the capability shortfalls. Modeling and simulation is used to project improvements to existing systems and to identify and characterize capability shortfalls. M&S can also determine the impact of emerging technological opportunities on gross system performance. What force effectiveness can be realized?

The shortfall may result in a need to: (1) change doctrine, tactics, training, or organization; (2) modify an existing system; or (3) develop a new operational capability [4]

7.1.2. Capability with New System or Operating Concept

In addition to upgrades and new systems, M&S can explore the broad capability implications of alternative tactics and operational concepts. M&S can predict force effectiveness in future scenarios to determine what changes in operational concepts are indicated.

7.2. Concept Exploration Phase (Milestone 0)

During Mission Area Analysis, the Mission Element Needs Statement is approved. This initiates the first formal phase of the acquisition process, the concept exploration phase. The main role of modeling and simulation is to estimate the probable performance of the system which incorporates some new technology. Through the use of computer-aided design techniques and engineering analysis, the system design is subsequently refined and the likely effectiveness of the resultant system is studied by the use of detailed engagement models. A particularly good example of this kind of model is the Advanced Air-to-Air System Performance Evaluation model or AASPEM [5,6].

AASPEM is a simulation of the performance of proposed aircraft, missiles, or avionics in combat encounters controlled by a set of layered pilot decision logic tables. The overall model is comprised of many sub-models including five degree-of-freedom aircraft and

missile models, and offensive and defensive mission avionics models. Representative inputs include aircraft initial conditions (location, altitude, velocity), aircraft performance capability, weapons loads, aircraft signatures, sensor suites and their performance, data links, and decision logic for the rules of engagement. AASPEM has many available modes of operation: one-on-one within visual range, M on N (where M and N represent any two numbers) beyond visual range, pilot vs. pilot, pilot vs. computer, or computer vs. computer.

AASPEM has had a substantial impact on DoD acquisition. It was used by the Air Force Armament Division (now part of the Aeronautical Systems Center) to develop Pre-Planned Product Improvements for our Advanced Medium Range Air-to-Air Missile (AMRAAM) and to perform a future air-to-air missile concept analysis. Boeing used AASPEM to do Advanced Tactical Fighter trade-off studies and other users include General Dynamics, Northrop, Raytheon and Hughes.

Specific issues that arise in this phase are: (1) System performance. Does the capability improvement survive a closer look with more realism? Does the system need to be refined via computer-aided design and engineering analysis? (2) Force effectiveness. What about interfaces with related systems and equipment? M&S provides the opportunity to study local conflict and force interactions with more detailed measures of effectiveness. (3) Can the system be acquired and operated at a reasonable cost?

7.3. Demonstration and Validation Phase (Milestone 1)

If concept exploration generates an approved solution to the requirement, a program office is formed and the next phase of the acquisition process, demonstration and validation begins. In this phase, modeling and simulation are used to investigate whether the theoretical performance of a new system is likely to be achievable and affordable. This is done through studies which analyze trade-offs between performance factors such as speed, survivability, range and payload, and other factors such as supportability and cost. The primary result of these analyses is the determination of the likely mission effectiveness and cost -- the workability and affordability, if you will, of each alternative system design and operating option.

The balance of system parameters is important in this phase. Is the theoretical system performance improvement achievable? What system configuration best satisfies the requirement? M&S is used to trade performance, supportability, and cost factors as well as analyze the system's effectiveness for all design/operating options. Is the proposed system affordable? Are commensurate system reliability, availability, and maintainability gains forthcoming? If it is affordable, what is the difference in force effectiveness? Does the system make a difference?

7.4. Engineering and Manufacturing Development Phase (Milestone 2)

The role of modeling and simulation during the engineering and manufacturing development phase of the acquisition process tends to be focused on whether the new system will meet both performance specifications and reliability, maintainability and cost objectives established as a result of analysis done in previous acquisition phases. In this phase, we also use modeling and simulation to design production and operating economies into the new system and to simulate operating environments for developmental test and evaluation purposes.

Modeling and simulation play an important part here too by identifying areas needing tests of actual hardware, by providing the foundation for operational test planning, and by simulating operational environments difficult to achieve in peacetime. They are also used to study pre-planned product improvement initiatives and to explore alternative system employment modes.

The primary question here is "Does the system meet specification with respect to Performance, Reliability, Availability, and Maintainability?" In this phase the production and operating economies must be designed in and analyzed.

DT&E cannot always operate in a realistic environment. M&S can be used to simulate the operating environment for DT&E purposes. In addition, it will provide a basis for test planning and identify areas needing hardware test.

7.5. Production and Deployment Period (Milestone 3)

The primary issue of this phase is "Does the system satisfy the user?" Would the system function better with modified or added equipment? In this Phase, M&S can be used to study Preplanned Product Improvement and explore weapon system alternatives.

The Theater Command and Control Simulation Facility, or TACCSF, is an example of a complex simulation used during this phase [7,8,9,10,11,12,13]. It is basically a set of high-fidelity, real-time, man-in-the-loop simulators of the NATO Central Region Air Defense System. It ran on 16 32-computers operating in a parallel mode. It was capable of simulating a 2000x2000 nautical mile area with terrain masking features, 2,000 active aircraft, and 128 exercise participants.

TACCSF was used for Identify Friend, Foe, or Neutral (IFFN) Joint Test Facility (JTF) tests which showed clearly that complete situational awareness of an air/land battle is difficult to obtain. These initial tests also indicated a strong need for integrated Army/Air Force/NATO operational testing of the IFFN system.

8. MODELING AND SIMULATION IN THE OPERATIONAL ENVIRONMENT

In addition to the acquisition process, M&S can be used to explore deployment and employment alternatives or to define modifications to the existing weapon systems. As such, the final phase in the lifecycle of the system is operations and support. This phase is both a continuation of Phase III (Milestone IV) of the Acquisition process but also includes additional M&S activity specifically addressing operational issues.

9. REFERENCES

1. Sisti, Alex F., "Large-Scale Battlefield Simulation Using A Multi-Level Model Integration Methodology," RL-TR-92-69, Rome Laboratory, Air Force Systems Command, Griffiss Air Force Base, NY, April 1992.
2. J.S.Prezemieniecki, *Acquisition of Defense Systems*, AIAA Education Series, 1993.
3. Major Wallace B. Frank, Jr., USAF, "A Management Approach to the 80s," *Journal of Defense Systems Acquisition Management*, Volume 3, Number 4, Autumn 1980.
4. Air Force Electronic Combat Test Process Guide, AF Pamphlet 58-5, HQ USAF/TEP, November 1992.
5. Boeing Military Airplane Company, "Man-In-The-Loop Air Combat Simulation" (briefing), Seattle, WA, 1987.
6. "Tactical Aircraft Producers Face Diminishing Prospects," *Aviation Week & Space Technology*, 22 February 1988.
7. Col George Lippermeier, "IFFN: Solving the Identification Riddle," *Defense Electronics*, January 1988, pages 83-84 and 87-88.
8. Donald C. Latham, "Testing C3I Systems," *ITEA Journal of Test and Evaluation*, Volume VIII (1987), Number 2, pages 14-23.
9. James E. Haile, IFFN Test Director, "Use of the IFFN Testbed for Evaluating Air C2," *Proceedings of ITEA '87 Symposium "System Testing - Development Through Deployment"*, 10-12 November 1987, pages 103-106.
10. Dr. Donald S. Coram and Richard B. Mintz, "Using Simulation To Reduce Development Risk," *SIGNAL. Journal of the Armed Forces Communications and Electronic Association*, Volume 41, Number 11, July 1987, pages 51-53, 55.
11. Charles Zuhoski, "C3I Modeling: Can We Do Better?," *SIGNAL. Journal of the Armed Forces Communications and Electronic Association*, Volume 41, Number 11, July 1987, pages 57-60.
12. "NSIA Study, Modeling & Simulation In Support of C3, Phase I, Executive Summary and Report," *National Industrial Security Association*, 9 September 1986.
13. "Simulation and Modeling Study Phase II, Final Report," *NSIA C3I Committee*, February 1987.

APPENDIX 2

HIERARCHICAL SIMULATION

1. CHAPTER OUTLINE

1. Chapter Outline.....	A2-1
2. Introduction.....	A2-1
3. Hierarchical Construction.....	A2-2
4. Application to Metamodeling	A2-3
5. References	A2-4

2. INTRODUCTION

The ability of a metamodel to support a hierarchical simulation objective is based on the concept of modular construction of models [1]. Given two models, if the model description is in the proper form, then we can create a new model by specifying how the input and output ports are connected. This allows modules (models) to be connected by an operation called coupling. If A and B are coupled together, then we have a new model, AB, which is a coupled model which is once again in a modular form.

In this sense, modularity means the description of a model in such a way that it has a recognized input and output through which all interaction is accomplished. The ability to couple the models is called closure under coupling and it enables the hierarchical construction of models.

Elements of model bases that are closed under coupling consist of both atomic and coupled models, each of which is called a component. While modular discrete event models still require specification of inputs and outputs, they must accommodate the fact that events determine the dynamics of the models. These events are both external and internal. The external events arrive at the input port and are processed by the model. Internal events come from within the models, can change the state of the model, and will impact the processing of the input.

3. HIERARCHICAL CONSTRUCTION

Each atomic model has three parts to its description:

1. The input-output specification giving the input and output ports their ranges.
2. Static structure giving the state and auxiliary variables and ranges.
3. Dynamic structure which provides the external and internal transition specification.

A coupled model has a different description

1. The input-output specification.
2. Names of the components that are coupled together.
3. Coupling specification.

Hierarchical construction, made possible by the successive coupling of larger and larger components, goes beyond standard object-oriented programming. Model descriptions must be converted into a class specification. A class specification is a template for generating identical instances of the same model along with a convention for naming the different instances of the same model.

The structure and components of a hierarchical, modular model are portrayed by a composition tree. Generalizing the composition to represent a family of models results in a system entity structure. In this structure, there can be several possible models to represent it. The decomposition of the structure is an aspect since there may be several possible decompositions for a given entity.

The entity structure/model base combination provides a unifying description of knowledge consistent with system theoretic insights. System theory distinguishes between structure (constitution of the system) and behavior (outer manifestation). Knowledge is represented in the decomposition, coupling, and taxonomies (class definitions). Behaviors (causal relationships) are integrated into the models.

Synthesis of Models. The entity structure and the model base combine to facilitate model construction. The model base contains files for the various model classes. Model construction consists of two passes. First, there is a top-down pruning of the entity structure to identify the desired components in the model base. This is followed by a bottom-up synthesis to construct the new model. Elements of selected classes are coupled together following the coupling specification.

4. APPLICATION TO METAMODELING

Figure A2.1.1 Depicts the metamodeling process [2]. Metamodels of these models can be constructed in a modular fashion in the same manner as the models. Atomic metamodels are the metamodels of elements of simulations while coupled metamodels are the combination of these elements to form the simulation or simulation network.

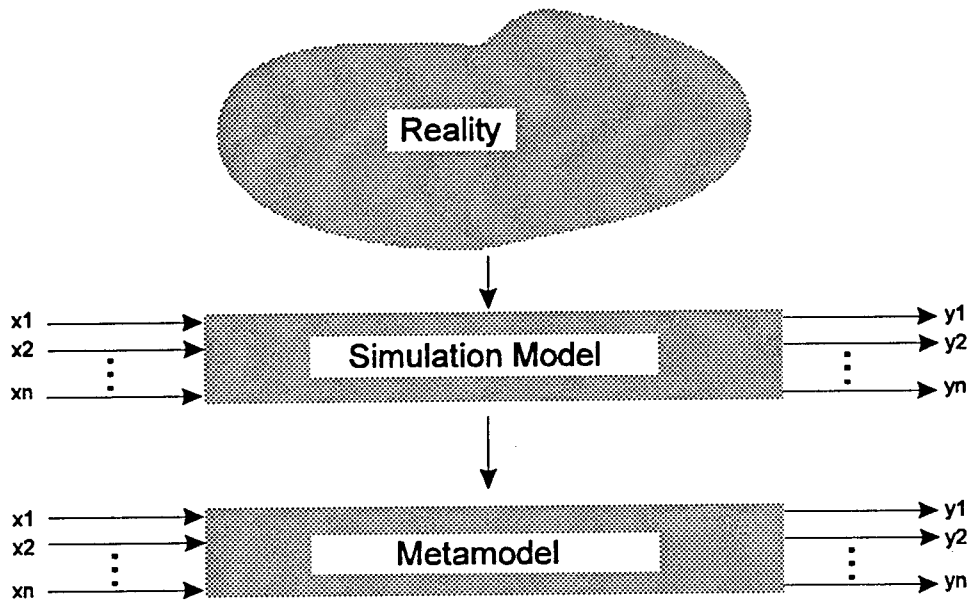


Figure A2.1.2 Metamodeling Process.

5. REFERENCES

1. B.P.Zeigler, "Hierarchical Modular Modeling/Knowledge Representation," *Proc. 1986 Winter Simulation Conf.*, p.p. 129-137, 1986.
2. U. Ilgenstein, R.G. Sargent, "An Investigation into Metamodeling," *CASE Center Technical Report No. 9405*, The New York Center for Advanced Technology in Computer Applications and Software Engineering, Syracuse University, NY, March 1994.

APPENDIX 3

ADAPTIVE SIMULATED ANNEALING

1. CHAPTER OUTLINE

1. Chapter Outline.....	A3-1
2. Introduction.....	A3-1
3. Metropolis Algorithm.....	A3-2
4. Statistical Mechanics.....	A3-3
4.1. Ensembles and Distribution Functions.....	A3-4
4.2. Entropy.....	A3-5
4.3. Entropy and Ensembles.....	A3-6
5. Information Theory.....	A3-7
5.1. Entropy for Equilibrium States.....	A3-7
6. Simulated Annealing.....	A3-8
6.1. Boltzmann Annealing.....	A3-8
6.2. Adaptive Simulated Annealing (ASA)}.....	A3-10
7. Summary.....	A3-12
8. References.....	A3-13

2. INTRODUCTION

One of the more significant problems of modeling simulations with reduced order (more abstract) models is the stochastic nature of the process and resulting distribution of the estimated coefficients. Using statistical mechanical theories, an optimization technique called "simulated annealing" provides a new option to directly process nonlinear, discontinuous, stochastic functions [1].

The techniques of system identification are based on a large body of research on estimation and control. Most of the solutions, however, have been restricted to the analysis and control of linear systems. In this case, there are rather elegant closed-form solutions that provide mathematically optimal results. Extensions to nonlinear systems have met with mixed results since nonlinear systems exhibit incredibly rich behavior and rarely admit closed-form solutions [2].

For control of nonlinear systems there are a number of robust linearization techniques. The introduction of the dynamic compensator, or controller can, through feedback linearization, insure that the linear approximations remain valid. The identification problem, however, does not allow modification of plant dynamics to maintain linear assumptions. Other techniques such as small signal linearization, linearization through numerical differentiation, or a weighted combination of central difference estimates from perturbation analysis are required. Unfortunately, these techniques either restrict the

region where the linearization is applicable or lose the true nature of the interactions in the linearization.

The objective, then, is to use a true nonlinear, stochastic representation of the system for identification. Simulated annealing provides a possible method. Given data and a cost function, it will globally optimize that function. As a combinatorial technique, it is not as satisfying as a closed-form solution or general results. However, at the present time, it may be the only method available to solve certain problems.

It should be noted that there is a similarity between simulated annealing in identification and *Dynamic Programming* in control. (Much of optimal control is based on *The Principle of Optimality* which was implemented in 1957 by Bellman via a procedure called *Dynamic Programming* [3].) Both are general numerical techniques that can provide reasonable solutions until more powerful methods are available to solve special classes of problems. However, in both cases, it is not possible to blindly apply the techniques to your problem and get a solution.

Like *Dynamic Programming*, simulated annealing requires special precautions in the setup of the problem. Without properly bounding the problem, *Dynamic Programming* suffered from the "curse of dimensionality" where the system became too large to solve. Likewise, simulated annealing will only optimize a function in a reasonable time if it is initialized properly. The relationship between temperature, entropy, and the state probability distribution functions should be understood.

The remainder of this appendix is organized to give the background required to apply simulated annealing to identify systems. Proper initialization requires an understanding of the Metropolis algorithm (Section 2), the statistical mechanics (Section 3) behind the technique (since it is based on the physical annealing process), and the connection to information theory for use in identification (Section 4). This background is then used in the discussion and comparison of simulated annealing codes (Section 5).

3. METROPOLIS ALGORITHM

The Metropolis algorithm is a combinatorial technique that can efficiently minimize a discrete objective function. It has attracted significant attention as suitable for problems of very large scale. The objective function is not simply the N dimensional space of continuously variable parameters, it is a very large discrete configuration space that cannot be exhaustively explored. Also, since the set is discrete, there is no definition of "direction" to be used in the search.

The Metropolis algorithm was the first method that relied on simulating the annealing process. Simulated annealing, as its name suggests, is related to thermodynamic annealing that identifies the way metals cool and anneal, or the way liquids freeze and crystallize. At high temperatures, molecules move freely with respect to one another. If the liquid is cooled slowly, the molecules are able to line up and form a crystal that is completely ordered and at minimum energy.

If the liquid is cooled quickly or "quenched," it does not reach this state. Instead it ends up in a state having more energy. Annealing is slow cooling, allowing time for the *ordered distribution* of molecules as mobility is lost.

Given a system in thermal equilibrium at temperature T with a distribution of energy states E , the Boltzmann probability distribution, $P(E) \approx \exp(-E/kT)$, gives the probability of the system being in each energy state. The quantity k (Boltzmann's constant) is a constant of nature that relates temperature to energy. When the system was in a high energy state, which is less probable at lower temperatures, there is a higher probability that the system could get out of a local energy minimum to find a more global minimum.

These principles were incorporated in combinatorial problems [4,5]. Given an option, a system was assumed to change its configuration from energy E_1 to energy E_2 with probability $p = \exp(-(E_2 - E_1)/kT)$. If E_2 was less than E_1 , then the transition was assigned a probability of 1. This general scheme became known as the Metropolis algorithm.

The following is required to use the Metropolis algorithm:

- A description of possible system configurations;
- A random change (number) generator that provides options to the system;
- An objective function to minimize;
- A control parameter T (the analog of temperature) and an annealing schedule which determines the rate at which the temperature drops. It must also determine the number of random changes in configuration that are allowed before a reduction in temperature.

The method has several attractive features. First, it is not easily satisfied by achieving a local minimum. It will continue to test minima at depths near its temperature. Second, configuration decisions tends to progress in a logical order. As T is larger, greater energy differences are considered. As the temperature drops, the decisions become more permanent with smaller refinements considered.

4. STATISTICAL MECHANICS

While distribution functions can be defined for a particular element of a system, statistical mechanics deals with distribution functions for complete thermodynamic systems [6]. To introduce this, define a microstate for a system of N elements as the specification of the $6N$ position and momentum coordinates (as long as the variables are independent, these coordinates could be generalized to other system characteristics). The value of the distribution function for this microstate is the probability density that the system has these coordinate values. Geometrically speaking, an elementary region in this $6N$ -dimensional phase space represents a microstate of the system. The fraction of time that the system spends in that microstate is proportional to the value of distribution function corresponding to that thermodynamic (system) state. **Statistical mechanics develops methods for finding distribution functions that correspond to specific thermodynamic states.**

4.1. Ensembles and Distribution Functions

The distribution function (as opposed to a **value** of the distribution function) represents the probability density, not for one system, but for a collection of similar systems. Consider a large number of identical systems in the **same thermodynamic state** but each in a different possible microstate. This collection of systems is called an ensemble of systems, the ensemble elements corresponding to the specified microstate. The distribution function for the thermodynamic system measures the relative number of systems in the ensemble that are in a given microstate at any instant.

Assume that each system in the ensemble has $\phi = 3N$ coordinates (degrees of freedom) q^i (position) and $\phi = 3N$ coordinates p^i (momenta). From classical mechanics, the total energy for the system is the Hamiltonian function $H(p, q)$ which is the sum of the elemental

Hamiltonians $H(p, q) = \sum_{i=1}^{\phi} \frac{1}{2m^i} (p^i / h^i)^2 + \sum_{i=1}^{\phi} \rho(q^i)$, where m^i is the mass and ρ is the

potential energy, and h^i is a scale factor (if required) for the curvilinear coordinates. Since each system in the ensemble is independent from all others, the Hamiltonian will be independent of time and the values of the q 's and p 's at a given instant determine the position in the 2ϕ phase space. The motion of the system point in phase space is determined by the equations of motion of the system. In this case (of momentum and position) it is Hamilton's equations:

$$\dot{q}^i = (\partial H / \partial p^i) \quad \dot{p}^i = -(\partial H / \partial q^i) \quad i = 1, 2, \dots, \phi$$

The ensemble of systems can thus be represented as the aggregate of these system points in phase space, each point moving according to the equations of motion above. The distribution function for the ensemble of systems is:

$$f(q, p) = f(q^1, q^2, \dots, q^\phi, p^1, p^2, \dots, p^\phi)$$

where $f(q, p)dV_q dV_p$ is the probability that a system chosen from the ensemble is within the element $dV_q dV_p$ of phase space. Note that since p^i is independent of direction and q^i is defined along the basis of the space, the element of momentum space is a hypersphere and an element of position space is a hypercube.

If the ensemble state is an equilibrium state, the density of points in any specified region will be constant, as many system points will enter the region as will leave it. If the thermodynamic state of the ensemble is not an equilibrium state $f(q, p)$ will be a function of time. Since each point in phase space represents an individual system, and since all of the systems in the ensemble are independent, the equation of continuity in phase space represents the fact that as the points move about in phase space, no point either appears or disappears:

$$\frac{\partial f}{\partial t} + \sum_{i=1}^{\Phi} \frac{\partial}{\partial q^i} (\dot{q}^i f) + \sum_{i=1}^{\Phi} \frac{\partial}{\partial p^i} (\dot{p}^i f) = 0 \quad (13-4)$$

Since each system in the ensemble obeys Hamilton's principle, the continuity equation becomes:

$$\frac{\partial f}{\partial t} + \sum_{i=1}^{\Phi} \left[\frac{\partial}{\partial q^i} \left(f \frac{\partial H}{\partial p^i} \right) - \frac{\partial}{\partial p^i} \left(f \frac{\partial H}{\partial q^i} \right) \right] = 0$$

since

$$\frac{\partial H}{\partial p^i} \frac{\partial f}{\partial q^i} = \frac{\partial}{\partial q^i} \left(f \frac{\partial H}{\partial p^i} \right) - f \left(\frac{\partial^2 H}{\partial q^i \partial p^i} \right)$$

we have

$$\frac{\partial f}{\partial t} + \sum_{i=1}^{\Phi} \left[\left(\frac{\partial H}{\partial p^i} \frac{\partial f}{\partial q^i} \right) - \left(\frac{\partial H}{\partial q^i} \frac{\partial f}{\partial p^i} \right) \right] = 0$$

or (Liouville's theorem)

$$\frac{\partial f}{\partial t} + \sum_{i=1}^{\Phi} \left[\left(\dot{q}^i \frac{\partial f}{\partial q^i} \right) - \left(\dot{p}^i \frac{\partial f}{\partial p^i} \right) \right] \equiv \frac{df}{dt} = \frac{\partial f}{\partial t} + \mathbf{U} \cdot \text{grad}(f) = 0$$

4.2. Entropy

Combining the Boltzman probability distribution with the equations describing the dynamics of the distribution functions, we have related the distribution of system states to the thermodynamic state of the ensemble. Another measure of the state of the ensemble of systems is entropy. The concept of entropy stems from the second law of thermodynamics that states that the spontaneous tendency of a system to go toward equilibrium cannot be reversed without changing some organized energy (work) into disorganized energy (heat).

The entropy of the system $S(x,y)$ is an extensive variable (proportional to the size of the system) equal to the integral of the perfect differential $dS = \frac{dQ}{T}$. Where a perfect differential is any differential that integrates to zero around any closed path, dQ is the heat absorbed by the system, and T is the thermodynamic temperature. The entropy is the

¹From Eq(13-4)

extensive variable that pairs with the intensive variable (independent of the size of the system) Q .

The fact that the perfect differential integrates to zero around a closed path, implies that $dS = \frac{dQ}{T}$ only holds for reversible processes (see, for example, [6] on a discussion of Carnot cycles). An irreversible process is a spontaneous process going automatically in one direction only. If the process is irreversible, the integral of $\frac{dQ}{T}$ is different from zero: since it must be less efficient than a reversible process, the second law of thermodynamics requires it to be less than zero. Also, since TdS has the dimensions of energy and equals that amount of heat given to the system, an irreversible process will not have all of that heat available for use so that $TdS > dQ$.

Consequently, since $\oint \left(\frac{dQ}{T} \right) \leq 0$, and $dS = \frac{dQ}{T}$ when the process is reversible, we have $dS \geq \frac{dQ}{T}$. The equalities hold for reversible processes and the inequalities hold for irreversible processes. Note: dQ is the heat absorbed by the system, therefore, $\oint \left(\frac{dQ}{T} \right) < 0$ is heat (disorganized energy) removed from (lost to) the system. Therefore, for an irreversible process $\oint dS = S_f - S_i > \oint \left(\frac{dQ}{T} \right) = 0$, so that entropy always increases.

Entropy is a measure of the unavailability of heat energy. The entropy of a given quantity of heat at low temperature is greater than the entropy of the same quantity of heat at a higher temperature. Since heat at a lower temperature has less potential to do work (organized energy), an increase in entropy is an increase in disorder. Irreversible (spontaneous) processes increase disorder, increase the amount of low temperature heat, and thus increase the entropy of the universe. Reversible processes, on the other hand, simply transfer entropy from one body to another, keeping entropy constant.

4.3. Entropy and Ensembles

The basic postulate that relates the distribution function f_v to the thermodynamic properties that the ensemble represents is given by the equation:

$$S = -k \sum_v f_v \ln(f_v) \quad \sum_v f_v = 1$$

where S is the entropy of the system. This equation is consistent with our concept of entropy. An ensemble that is in a single state with probability 1, has zero entropy. There is no disorder. A system that has an equal probability of being in any of N states has entropy $S = -k \sum_{v=1}^N \frac{1}{N} \ln\left(\frac{1}{N}\right) = k \ln(N)$ which increases as N increases. We do not have

any information as to the actual state of the system. Consequently, **disorder implies a lack of information.**

5. INFORMATION THEORY

We have implied a relationship between information and entropy. We have developed relationships between entropy and an ensemble of systems and between the ensemble and distribution function that will be used in the annealing algorithm. We will now use information theory to develop an explicit relationship between information and entropy that is exploited in simulated annealing.

If there are N possible messages, and if the probability that the i^{th} message will be sent is f_i , then the information that would be obtained if the message i were received must be a function $I(f_i)$, which increases as the term $1/f_i$ increases. The less likely the message, the greater the information conveyed if the message is sent.

Assume that the information from a set of message is additive $I(f_1 f_2) = I(f_1) + I(f_2)$. The probability of multiple messages is $f_1 f_2 \dots f_n$. Therefore the function $I(f_i)$ must be a logarithmic function of f , $I(f_i) = -K \ln(f_i)$.

Consequently, with reference to the basic postulate, entropy is proportional to the lack of detailed information about the system. Consider now an ensemble corresponding to a thermodynamic state f_v . To find out the microstate of the ensemble, we would have to measure its state. Or, we could use the expected amount of information we would obtain as a measure of our present lack of knowledge of the system (i.e., the system's disorder). The expected amount of information we would obtain is the weighted mean of $-K \ln(f_v)$ over all of the quantum states in the ensemble: $-K \sum_v f_v \ln(f_v) = -(K/k)S$.

5.1. Entropy for Equilibrium States

Now that we have defined the entropy of an ensemble, we will use:

$$S = -k \sum_v f_v \ln(f_v) \quad \sum_v f_v = 1$$

to find the distribution function for the thermodynamic states. In an isolated system, S tends to increase until it is as large as it can be subject to the restrictions on the system. (Irreversible - spontaneous - processes move in one direction only and increase entropy.)

Assume that the number of microstates represented by f_v is finite. Also assume that the derivative of S with respect to each independent f be zero (equilibrium). Note: In optimization, a maximum can be met when the derivative is zero or at either boundary condition. Therefore the problem is to determine the value of each f_v so that:

$$S = -k \sum_v^W f_v \ln(f_v) \text{ is maximum - subject to } \sum_{v=1}^W f_v = 1$$

Using the calculus of variations, this equation can be solved by maximizing $S(f_1, \dots, f_W) + \alpha_0 \sum_{v=1}^W f_v$ with α_0 determined so that $\sum_{v=1}^W f_v = 1$. Setting each partial equal to zero gives:

$$0 = \frac{\partial}{\partial f_k} \left[\alpha_0 \sum_{v=1}^W f_v - k \sum_{v=1}^W f_v \ln(f_v) \right] = \alpha_0 - k \ln(f_k) - k$$

or

$$f_k = \exp[(\alpha_0 / k) - 1].$$

Since neither α_0 nor k depend on K , all f 's are equal. Thus α_0 is computed from the requirement $\sum_{v=1}^W f_v = 1$. This gives $f_v = (1/W)$ so that:

$$S = -k \sum_{v=1}^W (1/W) \ln(1/W) = k \ln W.$$

6. SIMULATED ANNEALING

With an established connection between the information available for identification of the parameter distributions and a single number (representing the entropy of the system) we can proceed with an explanation of the simulated annealing algorithm.

The discussion on the Metropolis algorithm highlighted the basics of the simulated annealing algorithm. This original method is referred to as Boltzmann annealing.

6.1. Boltzmann Annealing

The method combines three functional relationships: A **state generating function** that reflects the probability density of the N parameters $\mathbf{x} = \{x^i; i = 1, \dots, N\}$; an **acceptance probability** for accepting a new value of the cost function given the previous value; and a **schedule of "annealing"** the "temperature" which changes the fluctuations in either or both of the two previous densities.

For a state generating function, consider a set of states $\{\mathbf{x}\}$. each with energy $e(\mathbf{x})$ the sum equaling the total energy E . With this set of states, there is a probability distribution $p(\mathbf{x})$, and an energy distribution per state $d(e(\mathbf{x}))$. Therefore:

$$\sum_{\mathbf{x}} p(\mathbf{x}) d\{e(\mathbf{x})\} = E$$

Maximizing the entropy $S = -\sum_x p(x) \ln \frac{p(x)}{p(\bar{x})} = E$ (where \bar{x} represents the reference state) using Lagrange multipliers to constrain the energy to the average value of T , leads to the Gibbs distribution [7]:

$$g(x) = \frac{1}{\sum_x \exp[-H(x)/T]} \exp[-H(x)/T]$$

where H is the Hamiltonian operator as the energy function. Other generating functions can be derived using the same procedures with different constraints. For example, Gauss-Markov systems can use:

$$g(\Delta x) = \frac{1}{(2\pi T)^{N/2}} \exp[-\Delta x^2 / 2T]$$

The acceptance probability is based on the probability of obtaining a new state at update $k+1$ with "energy" E_{k+1} relative to a previous state k with "energy" E_k :

$$\begin{aligned} h(\Delta E) &= \frac{\exp[-E_{k+1}/T]}{\exp[-E_{k+1}/T] + \exp[-E_k/T]} \\ &\approx \exp[-\Delta E/T] \end{aligned}$$

where ΔE is the "energy" difference between the cost functions.

The "annealing schedule" is selected to "statistically" insure that a global minimum of the state generating function is obtained. The rate of cooling must satisfy sufficient conditions for a (weak) ergodic search. (Ergodicity refers to the ability to interchange time and ensemble averages.) For the Gauss-Markov generating function, for example, a global minimum of $E(x)$ will be obtained if T does not "cool" any faster than

$$T(k) = \frac{T_0}{\ln(k)}$$

when T_0 is "large enough" [7].

In general, the annealing schedule for each application requires experimentation. The basic rule is to choose a starting value of the parameter T_0 that is considerably larger than the largest ΔE normally encountered.

Being a quite general combinatorial technique, a number of modifications have been made to Boltzmann annealing. In simulated quenching, the temperature schedule is accelerated at the risk of not obtaining a global minima. Mean-field annealing is a quenching algorithm that searches deterministic most likely trajectories rather than performing a fully

stochastic search. This technique is useful for quadratic energy functions when the mean function is a good approximation to the stochastic cost function.

While the above modifications have sacrificed the ergodic nature of the search, there have been modifications that have maintained this feature. Fast annealing uses the Cauchy distribution in place of the Boltzmann form. Adaptive simulated annealing explicitly allows for different time-dependencies in the annealing schedule of the different parameters. The generating functions for Boltzmann and fast annealing do not allow different annealing schedules for different parameters.

6.2. Adaptive Simulated Annealing (ASA)}

Adaptive simulated annealing was once titled "Very Fast Simulated Reannealing (VFSR)" [8]. Explicitly defined in N -dimensional space, the generating probability density function for adaptive simulated annealing considers each parameter individually. The range for the k^{th} update of parameter i is $p_k^i \in [A_i, B_i]$. This parameter is updated with the random variable x^i by:

$$p_{k+1}^i = p_k^i + x^i(B_i - A_i)$$

where $x^i \in [-1, 1]$.

This update is generated using a distribution defined by the product of distributions for each parameter, $g^i(x^i; T_i)$, in terms of random variables $x^i \in [-1, 1]$.

The generating probability density function at temperature T for the vector of random variables \mathbf{x} is:

$$g_T(\mathbf{x}) = \prod_{i=1}^N \frac{1}{2(|x^i| + T_i) \ln(1 + 1/T_i)} \equiv \prod_{i=1}^N g_T^i(x^i)$$

The cumulative probability distribution, $G_T(\mathbf{x})$ of this density is:

$$\begin{aligned} G_T(\mathbf{x}) &= \int_{-1}^{x^1} \cdots \int_{-1}^{x^N} g_T(\mathbf{x}') dx'^1 \cdots dx'^N \\ &\equiv \prod_{i=1}^N G_T^i(x^i) \end{aligned}$$

$$G_T^i(x^i) = \frac{1}{2} + \frac{\text{sgn}(x^i)}{2} \frac{\ln[1 + |x^i|/T_i]}{\ln[1 + 1/T_i]}$$

Therefore, the random variable x^i is generated from $u^i \in U[0, 1]$ by:

$$x^i = \text{sgn}(u^i - \frac{1}{2}) T_i [(1 + 1/T_i)^{|2u^i - 1|} - 1]$$

The acceptance probability density function uses a Boltzmann test. At each annealing time $k+1$, the cost functions, $C(p_{k+1}) - C(p_k)$, are compared using a uniform random generator, U in $[-1,1]$. If

$$\exp[-(C(p_{k+1}) - C(p_k)) / T_{\text{cost}}] > U$$

where T_{cost} is the "temperature" used for this test, then the new point is accepted as the new saved point for the next iteration. Otherwise, the last saved point is retained.

Starting with temperature T_{i0} , the annealing temperature schedule (for the parameters) at the annealing time k for this generating function is:

$$T_i(k_i) = T_{oi} \exp[-c_i k_i^{1/N}]$$

The parameter c_i is controlled so that $T_{fi} = T_{oi} \exp[-m_i]$ when $k_f = \exp[n_i]$ so that

$$c_i = m_i \exp[-n_i / N]$$

where m_i and n_i are "free" parameters used to tune ASA for specific problems.

The annealing schedule for the cost temperature is developed similarly to the parameter temperatures. However, the index for reannealing the cost function, T_{cost} is determined by the number of accepted points, instead of the number of generated points as used for the parameters. This choice was made because the Boltzmann acceptance criteria uses an exponential distribution which is not as fat-tailed as the ASA distribution used for the parameters.

A multi-dimensional search should deal with the changing sensitivities of the different parameters. This is accomplished in ASA by periodic reannealing (rescaling the annealing time k) of the generating function to "stretch out" the range over which the relatively insensitive parameters are being searched.

The sensitivity of the parameters s_i is calculated at the current minimum value of the cost function C via $s_i = \partial C / \partial p^i$. The maximum sensitivity s_{max} is used with each parameter:

$$T'_{ik} = T_{ik} (s_{\max} / s_i)$$

$$k_i \rightarrow k'_i = \left(\frac{\ln[T_{i0} / T'_{ik}]}{c_i} \right)^N$$

with T_{i0} set to unity to begin the search.

The acceptance temperature is similarly rescaled.

7. SUMMARY

Compared to other optimization techniques, simulated annealing is not efficient. If the structure of the problem is well known, one of these other optimization techniques may be more appropriate. If the structure of the system is not well known, and especially if there are complex constraints, then (properly initialized) ASA provides a good search technique for a global optimum. Some of the advantages of ASA are:

- The algorithm can process cost functions possessing quite arbitrary degrees of nonlinearities, discontinuities, and stochasticity;
- ASA provides a global minimum in parameter space more certain than with regression fitting;
- All parameters, including the noise, are simultaneously and equally treated;
- Boundary conditions can be explicitly included for each parameter;
- ASA can handle higher order models.

8. REFERENCES

1. A. L. Ingber, Draft of "Statistical Mechanical Aids to Calculating Term Structure Models," J. Phys. Rev. Vol 42, 1990.
2. E. Bradley and F. Zhao, "Phase-Space Control Design," *IEEE Control System Magazine*, April 1993.
3. R. Bellman, *Dynamic Programming*, Princeton University Press, Princeton, N.J., 1957.
4. N Metropolis et al, *Journal of Chemical Physics*, Vol. 21, p. 1087, 1953.
5. Press et al, *Numerical Recipes*, Cambridge University Press, 1986.
6. P.M. Morse, *Thermal Physics*, W.A. Benjamin, NY, 1965.
7. L. Ingber, "Simulated Annealing: Practice versus Theory," reprint from *Journal Mathl. Comput. Modeling*, December 1993.
8. L. Ingber, "Statistical Mechanics of Combat and Extensions," reprint from *Toward a Science of Command, Control, and Communications*, AIAA, December 1993.

***MISSION
OF
ROME LABORATORY***

Mission. The mission of Rome Laboratory is to advance the science and technologies of command, control, communications and intelligence and to transition them into systems to meet customer needs. To achieve this, Rome Lab:

- a. Conducts vigorous research, development and test programs in all applicable technologies;
- b. Transitions technology to current and future systems to improve operational capability, readiness, and supportability;
- c. Provides a full range of technical support to Air Force Materiel Command product centers and other Air Force organizations;
- d. Promotes transfer of technology to the private sector;
- e. Maintains leading edge technological expertise in the areas of surveillance, communications, command and control, intelligence, reliability science, electro-magnetic technology, photonics, signal processing, and computational science.

The thrust areas of technical competence include: Surveillance, Communications, Command and Control, Intelligence, Signal Processing, Computer Science and Technology, Electromagnetic Technology, Photonics and Reliability Sciences.